



# **HTNG Product Distribution Availability Specification Version 4.0**

**21 October 2013**

## About HTNG

Hotel Technology Next Generation (HTNG) is a non-profit association with a mission to foster, through collaboration and partnership, the development of next-generation systems and solutions that will enable hoteliers and their technology vendors to do business globally in the 21st century. HTNG is recognized as the leading voice of the global hotel community, articulating the technology requirements of hotel companies of all sizes to the vendor community. HTNG facilitates the development of technology models for hospitality that will foster innovation, improve the guest experience, increase the effectiveness and efficiency of hotels, and create a healthy ecosystem of technology suppliers.

Copyright 2013, Hotel Technology Next Generation

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the copyright owner.

For any software code contained within this specification, permission is hereby granted, free-of-charge, to any person obtaining a copy of this specification (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the above copyright notice and this permission notice being included in all copies or substantial portions of the Software.

Manufacturers and software providers shall not claim compliance with portions of the requirements of any HTNG specification or standard, and shall not use the HTNG name or the name of the specification or standard in any statements about their respective product(s) unless the product(s) is (are) certified as compliant to the specification or standard.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES, OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF, OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Permission is granted for implementers to use the names, labels, etc. contained within the specification. The intent of publication of the specification is to encourage implementations of the specification.

This specification has not been verified for avoidance of possible third-party proprietary rights. In implementing this specification, usual procedures to ensure the respect of possible third-party intellectual property rights should be followed. Visit <http://htng.org/ip-claims> to view third-party claims that have been disclosed to HTNG. HTNG offers no opinion as to whether claims listed on this site may apply to portions of this specification.

The names Hotel Technology Next Generation and HTNG, and logos depicting these names, are trademarks of Hotel Technology Next Generation. Permission is granted for implementers to use the aforementioned names in technical documentation for the purpose of acknowledging the copyright and including the notice required above. All other use of the aforementioned names and logos requires the permission of Hotel Technology Next Generation, either in written form or as explicitly permitted for the organization's members through the current terms and conditions of membership.

# Table of contents

21 OCTOBER 2013 .....	1
<b>1 THIS SPECIFICATION AT A GLANCE.....</b>	<b>5</b>
<b>2 DOCUMENT INFORMATION .....</b>	<b>6</b>
2.1 DOCUMENT HISTORY .....	6
2.2 DOCUMENT PURPOSE .....	7
2.3 SCOPE .....	7
2.4 RELATIONSHIP TO OTHER STANDARDS.....	7
2.5 USEFUL RESOURCES.....	7
2.6 AUDIENCE.....	8
2.7 OVERVIEW .....	8
2.8 FURTHER CONSIDERATIONS.....	8
<b>3 COMPONENT SCENARIOS.....</b>	<b>9</b>
3.1 UPDATE AVAILABILITY .....	10
3.1.1 <i>Overview</i> .....	10
3.1.2 <i>Roles</i> .....	10
3.1.3 <i>Use Case</i> .....	11
3.1.4 <i>Message Flows</i> .....	12
3.1.5 <i>Update Room Type Availability</i> .....	12
3.1.6 <i>Update Rate Availability</i> .....	13
3.1.7 <i>Update Room/Rate Availability</i> .....	14
3.1.8 <i>Update Segment Availability</i> .....	15
3.1.9 <i>Update Segment/Room Availability</i> .....	16
3.1.10 <i>Update House Availability</i> .....	18
3.1.11 <i>Update Hurdle Availability</i> .....	19
3.1.12 <i>Update Best Available Rates</i> .....	20
3.2 GET AVAILABILITY .....	21
3.2.1 <i>Overview</i> .....	21
3.2.2 <i>Roles</i> .....	21
3.2.3 <i>Use Case</i> .....	22
3.2.4 <i>Message Flows</i> .....	22
3.2.5 <i>Get Availability by Date Range</i> .....	23
3.2.6 <i>Get Availability by Rate Plan</i> .....	24
3.2.7 <i>Get Availability by Room Type</i> .....	26
3.2.8 <i>Get Availability by Rate Plan/Room Type Combination</i> .....	27
3.2.9 <i>Get Best Available Rates</i> .....	28
3.2.10 <i>Get Rate Hurdles</i> .....	29
3.2.11 <i>Get Availability Restrictions</i> .....	30
3.2.12 <i>Get Length of Stay Restrictions</i> .....	32

3.3	UPDATE INVENTORY .....	33
3.3.1	Overview .....	33
3.3.2	Roles .....	33
3.3.3	Use Case .....	33
3.3.4	Message Flows .....	35
3.3.5	Update Hotel Inventory .....	35
3.3.6	Update Room Type Inventory .....	36
4	MESSAGES .....	37
4.1	UPDATE AVAILABILITY .....	37
4.1.1	Data Element Table – Update Availability Request .....	37
4.1.2	Data Element Table – Update Availability Response .....	40
4.2	GET AVAILABILITY .....	42
4.2.1	Data Element Table – Get Availability Request .....	42
4.2.2	Data Element Table – Get Availability Response .....	45
4.3	UPDATE INVENTORY .....	50
4.3.1	Data Element Table – Update Inventory Request .....	50
4.3.2	Data Element Table – Update Inventory Response .....	51
5	APPENDICES .....	53
5.1	GLOSSARY OF TERMS .....	53
5.2	IMPLEMENTATION NOTES .....	53
5.2.1	Errors and Warnings Usage .....	53
5.2.2	Min/Max LOS .....	53
5.3	REFERENCED DOCUMENTS .....	54

## 1 This Specification at a Glance

This specification details a set of core services and a schema to be used in the exchange of Availability and Inventory information between systems.

The Availability use cases covered in this document include the following:

- Update Availability
  - Update Room Type Availability
  - Update Rate Availability
  - Update Room/Rate Availability
  - Update Segment Availability
  - Update Segment/Room Availability
  - Update House Availability
  - Update Hurdle Availability
  - Update Best Available Rates
- Get Availability
  - Get Availability by Date Range
  - Get Availability by Rate Plan
  - Get Availability by Room Type
  - Get Availability by Rate Plan/Room Type Combination
  - Get Best Available Rates
  - Get Rate Hurdles
  - Get Availability Restrictions
  - Get Length of Stay Restrictions
- Update Inventory
  - Update Hotel Inventory
  - Update Room Type Inventory

This specification is designed to be implemented in any system that wishes to exchange Availability and/or Inventory messages, such as Open/Close, allocations, and minimum and maximum length of stay restrictions.

## 2 Document Information

### 2.1 Document History

Version	Date	Author	Comments
1.0.0	27 Dec 2005	Juan Gasparini	First draft posted for workgroup review
1.0.1	13 Feb 2006	Juan Gasparini	All/Editorial changes from workgroup review
2.0.0	20 July 2007	Kevin Smith	Revised draft for inclusion of BAR – posted for Workgroup review
2.0.1	07 Oct 2007	Kevin Smith	Re-draft of TaxInclusive change
2.0.2	11 Nov 2007	Martin Kirk	Reformat to HTNG House Style
2.0.3	12 Nov 2007	Martin Kirk	Minor corrections
2.2.0	20 Jul 2009	Lew Harasymiw	Changes to implementation of RestrictionStatus from previous version; updates to use cases
2.3.0	11 May 2012	Product Distribution (PD) Workgroup	Added AvailStatusMessage and updated ArrivalDateBased attributes
2.3.01–.03	12 Nov 2012	PD Workgroup	Updated scenarios
2.0.04–2.3.21	Dec 2012–Mar 2013	PD Workgroup	Updated messages
2.90	22 Mar 2013	Kylene Reese	Prepared for member review period
2.91	25 Mar 2013	PD Workgroup	Addressed member comments
2.95	10 Apr 2013	Kylene Reese	Prepared spec for Workgroup vote
3.0	19 Apr 2013	PD Workgroup	Separated the specification from larger, all-inclusive Product Distribution document
3.01	30 May 2013	Jay Rosamilia & Kylene Reese	Included Inventory scenarios
3.02 – .03	10 Jun 2013	PD Workgroup	Reviewed changes with the Workgroup and updated sub-scenarios; moved groups-specific info to the Groups spec
3.02.04	11 Jun 2013	Kylene Reese	Moved all groups information to the Groups spec
3.2.05	20 Jun 2013	Kylene Reese	Moved Error Handling Appendix to its own document

3.2.06	24 Jun 2013	PD Workgroup	Removed Get Inventory scenario
3.2.07– .10	Sep 2013	Jay Rosamilia	Updated data element tables
3.90	19 Sep 2013	Kylene Reese	Prepared document for member review
3.99	14 Oct 2013	Kylene Reese	Prepared document for workgroup vote
4.0	21 Oct 2013	Kylene Reese	Prepared the document for General Release

## 2.2 Document Purpose

This document defines the Hotel Technology Next Generation (HTNG) scenarios and business processes for exchanging availability and inventory messages based on OpenTravel Alliance messages. This document provides a framework for trading partners to define various attributes related to availability controls to be exchanged between systems.

## 2.3 Scope

This document defines a standard HTNG implementation of the OpenTravel Alliance messages for inventory, availability and restrictions.

## 2.4 Relationship to Other Standards

This specification and its supporting schemas leverage the existing OpenTravel Alliance methodology for message construction and draws upon data definitions common to several HTNG specifications as of October 2013.

Related specifications as of October 2013:

- Other HTNG Product Distribution specifications – outline of most recent versions available on [workgroup's wiki page](#)
- [OpenTravel Alliance Specifications](#)

## 2.5 Useful Resources

- [Implementing Web Services Using HTNG Specifications – A Quick Start Guide for Software Developers](#)
- HTNG Discussion Board – currently available at <http://www2.htng.org/discussion>
- [Error Handling Appendix](#) – This is a companion document containing mandatory and optional error codes. This document is relevant when referring to warnings or errors, which are reported in message responses.

## 2.6 Audience

This document is designed as a guide for project managers, programmers and analysts to gain detailed information needed to implement the distribution of availability and inventory information between systems.

## 2.7 Overview

The OpenTravel Alliance messages contain a large number of optional fields, and the same information can be represented in a number of different ways.

The intent of the specification is to recommend a minimum common standard to represent complex data. The intent is also to avoid having to pass the same information in more than one field, thus avoiding confusion.

Trading partners may agree prior to implementation to use additional fields for data not covered in the scenario, including TPA extensions as per the OpenTravel Alliance schemas.

## 2.8 Further Considerations

The scenarios outlined provide a common starting point for the definition of the messages exchanged and that the implementers of these messages will:

1. Add expansions as needed to enable exchange of additional information while retaining compliance with the OpenTravel Alliance specifications.
2. Report expansions deemed common to HTNG for consideration as additional scenarios.
3. Report any missing elements or attributes to OpenTravel Alliance for inclusion in a future specification release.



### 3 Component Scenarios

Partners will be responsible for creating their own Quality Assurance Test Scripts.

The intent of the HTNG scenarios is to recommend a minimum common denominator and clarify what fields should be used to transfer the data required. The main aim is to avoid having to pass the same information in more than one field, thus avoiding confusion.

Trading partners may agree prior to implementation to use additional fields for data not covered in the scenario, including TPA extensions as per the OpenTravel Alliance specifications schema.

#### A Note about required fields

In each of the scenarios below, sample request and response messages are provided. In each of these samples, **bold typeface** is used on some data values specific to a given scenario which differentiates it from other scenarios; these data values **MUST** be populated in addition to the required data as outlined in the corresponding Data Element Table.

#### Behavior

The process being facilitated is that of updating or getting availability and/or restrictions and/or inventory information between systems [generally, but not exclusively, a Property Management System (PMS) and a Central Reservation System (CRS)].

Based on the type of information that needs to be exchanged, trading partners will select the appropriate scenario and fill in the Extensible Markup Language (XML) according to the guidelines provided in the specific scenario.

XML will then be sent to the appropriate webservice, and the correct response will be used for the selected scenario. To facilitate the ease of troubleshooting, it is highly recommended that both the sending and receiving systems log communications.

#### Behavior Expected Prior to Transmission

The behavior expected from a sending system will be:

- Usage of the appropriate scenario for the type of information to be sent
- Translation of availability information into codes understood by the receiving system (if required)
- Provision of all fields classed as Mandatory in the scenario
- Validation of XML message format
- Transmission of the XML to appropriate URL set up by the receiving system using the appropriate identification criteria (SOAP user and password provided by the receiving system)

### Behavior Expected from Receiving System

Upon receipt of the XML message the behavior expected from the receiving system will be:

- Process XML received by the sending system
- Update (in the update scenario) or return (in the get scenario) hotel data with availability and/or restriction information contained in the XML sent by the sending system
- If update is not possible or if it is not possible to process the message: create an error response as defined in the scenario and transmit the response to the sending system
- Acknowledge successful/unsuccessful processing of the upload using the appropriate response message

### Behavior Expected Upon Processing by Receiving System

Once the initial request has been processed by the receiving system, the sending system will need to:

- Be able to evaluate error responses coming from the receiving system
- React to error responses as needed: by modifying the XML and/or ensuring correct mapping tables are used

## **3.1 Update Availability**

### **3.1.1 Overview**

This process describes the interaction between systems where notification of changes to availability must be communicated. The update availability scenario covers the pushing of availability settings from one system that defines them or provides a user an interface to define them or receives them from another system to another system with the ability to book or change a reservation.

This scenario includes any availability settings that can be pushed from one system to another. It does not include pulling availability settings.

### **3.1.2 Roles**

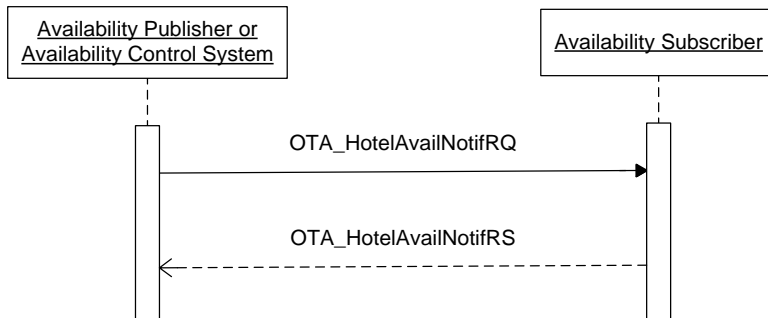
Role Name	Definition	Examples
Availability Publisher OR Availability Control System	A system that maintains availability and notifies Availability Subscriber(s) of changes.	<ul style="list-style-type: none"><li>• Central Reservation System</li><li>• Property Management System</li><li>• Channel Management System</li><li>• Yield Management System</li></ul>
Availability Subscriber	A system that requires notification of availability changes as they occur in Availability Publisher system.	<ul style="list-style-type: none"><li>• Central Reservation System</li><li>• Property Management System</li><li>• Channel Management System</li><li>• Revenue Management System</li><li>• Online Travel Agent</li></ul>

### 3.1.3 Use Case

Assumptions:	<ul style="list-style-type: none"><li>• Publisher/control system and subscriber have agreed which attributes will be transmitted and accepted.</li><li>• Publisher/control system and subscriber have agreed upon a notification model (either real-time or timed interval).</li></ul>
Pre-condition:	None
Trigger:	A change to availability occurs in the Availability Publisher's system.
Basic Course of Events:	<ol style="list-style-type: none"><li>1. Availability Publisher/Control System constructs OTA_HotelAvailNotifRQ containing the availability change(s).</li><li>2. Availability Publisher/Control System successfully transmits message payload.</li><li>3. Availability Subscriber receives the message payload.</li><li>4. Availability Subscriber processes OTA_HotelAvailNotifRQ.</li><li>5. Availability Subscriber acknowledges with the OTA_HotelAvailNotifRS indicating processing status of the message.</li><li>6. Availability Publisher/Control System may update their database to log what was last successfully processed by the Subscriber.</li></ol>
Post-condition:	Availability Subscriber system may update their log with what was last successfully processed.
Exception Path 1:	<ol style="list-style-type: none"><li>1. At step 3, the Availability Subscriber does not successfully receive the message.</li><li>2. Availability Subscriber does not return the OTA_HotelAvailNotifRS.</li><li>3. Availability Publisher/Control System may:<ol style="list-style-type: none"><li>a. Log the unsuccessful send.</li><li>b. Resend the update notification immediately.</li><li>c. Re-queue the update notification.</li><li>d. Wait until the next agreed interval to send an update notification.</li></ol></li></ol>
Exception Path 2:	<ol style="list-style-type: none"><li>1. At Step 4, the Availability Subscriber does not successfully process the message.</li><li>2. Availability Subscriber returns OTA_HotelAvailNotifRS with the appropriate error indicator.</li><li>3. Availability Publisher/Control System may:<ol style="list-style-type: none"><li>a. Log the response error.</li><li>b. Take action to correct the error and/or resend the transaction if necessary.</li><li>c. Re-queue the update notification.</li><li>d. Wait until the next agreed interval to send an update notification.</li></ol></li></ol>

Alternative Path:	If the Subscriber wishes to take on role of <i>Availability Requester</i> , it then may perform a <a href="#">Get Availability</a> .
-------------------	--

### 3.1.4 Message Flows



### 3.1.5 Update Room Type Availability

This defines the process between publisher and subscriber when a change to room type availability at a specific property has occurred and an updated status must be communicated to the subscriber. The message assumes a push model, with the publishing system pushing the data to the subscriber.

Specific to this message:

- MessageContentCode must have a value of “1”
- InvTypeCode must be populated
- At least one of the following MUST be implemented:
  - LengthsOfStay
  - BookingLimit
  - RestrictionStatus

#### 3.1.5.1 Sample Scenario for Update Room Type Availability with LengthsOfStay, RestrictionStatus and BookingLimit

PMS “ABC” transmits the following ROOM TYPE availability update to CRS “123,” requesting availability for the period from January 1, 2013 through January 14, 2013.

- Hotel Code: HXCAIZZ
- Change applies to DOW: FRI-SAT-SUN
- Room Type: A1K
- Number allotted: 10
- Status: On Request
- Min LOS: 2

### 3.1.5.1.1 Sample Request

```
<OTA_HotelAvailNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-08-01T09:30:47-05:00" EchoToken="b9d592ad-658f-4d2b-90c4-baa729df1a37" Target="Production" Version="1.001" TransactionIdentifier="938383" MessageContentCode="1">
  <!-- MessageContentCode=1 for Room Type Availability-->
  <AvailStatusMessages HotelCode="HXCAIZZ">
    <AvailStatusMessage BookingLimit="10" BookingLimitMessageType="SetLimit">
      <StatusApplicationControl Start="2013-01-01" End="2013-01-14"
    InvTypeCode="ALK" Mon="0" Tue="0" Weds="0" Thur="0" Fri="1" Sat="1" Sun="1"/>
      <LengthsOfStay>
        <LengthOfStay MinMaxMessageType="SetMinLOS" Time="2" TimeUnit="Day"/>
      </LengthsOfStay>
      <UniqueID Type="16" ID="1"/>
      <RestrictionStatus Restriction="Master" Status="OnRequest"/>
    </AvailStatusMessage>
  </AvailStatusMessages>
</OTA_HotelAvailNotifRQ>
```

### 3.1.5.1.2 Sample Response

```
<OTA_HotelAvailNotifRS xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-08-01T09:30:47-05:00" EchoToken="b9d592ad-658f-4d2b-90c4-baa729df1a37" Target="Production" Version="1.001" TransactionIdentifier="938383" MessageContentCode="1">
  <Success/>
  <Warnings>
    <Warning Type="3" Code="458" RecordID="1">
      <!-- Type 3 => Business Rule, Code 458 => Date Outside Inventory Period -->
    </Warning>
  </Warnings>
</OTA_HotelAvailNotifRS>
```

## 3.1.6 Update Rate Availability

This defines the process between publisher and subscriber when a change to rate availability at a specific property has occurred and an updated status must be communicated to the subscriber. The message assumes a push model, with the publishing system pushing the data to the subscriber.

Specific to this message:

- MessageContentCode must have a value of "2"
- RatePlanCode must be populated
- At least one of the following MUST be implemented:
  - LengthsOfStay
  - BookingLimit
  - RestrictionStatus

### 3.1.6.1 Sample Scenario for Update Rate Availability with RestrictionStatus and BookingLimit

PMS "ABC" transmits the following RATE availability update to CRS "123," updating availability for the period from January 1, 2013 through January 10, 2013.

- Hotel Code: HXCAIZZ

- Change applies to DOW: MON TO FRI
- Rate Plan: CR
- Number allotted: 15
- Status: Open

#### 3.1.6.1.1 Sample Request

```
<OTA_HotelAvailNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-08-01T09:30:47-05:00" EchoToken="31d7091a-8520-4571-b7ce-8e485c70ff7f" Target="Production" Version="1.001" TransactionIdentifier="938384" MessageContentCode="2">
  <!-- MessageContentCode=2 for Rate Availability-->
  <AvailStatusMessages HotelCode="HXCAIZZ">
    <AvailStatusMessage BookingLimit="15" BookingLimitMessageType="SetLimit">
      <StatusApplicationControl Start="2013-01-01" End="2013-01-10"
RatePlanCode="CR" Mon="1" Tue="1" Weds="1" Thur="1" Fri="1" Sat="0" Sun="0"/>
      <UniqueID Type="16" ID="1"/>
      <RestrictionStatus Restriction="Master" Status="Open"/>
    </AvailStatusMessage>
  </AvailStatusMessages>
</OTA_HotelAvailNotifRQ>
```

#### 3.1.6.1.2 Sample Response

```
<OTA_HotelAvailNotifRS xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-08-01T09:30:47-05:00" EchoToken="31d7091a-8520-4571-b7ce-8e485c70ff7f" Target="Production" Version="1.001" TransactionIdentifier="938384">
  <Success/>
</OTA_HotelAvailNotifRS>
```

### 3.1.7 Update Room/Rate Availability

Defines the process between publisher and subscriber when a change to a particular combination of a room type and rate plan at a specific property has occurred and an updated status must be communicated to the subscriber. The message assumes a push model, with the publishing system pushing the data to the subscriber.

Specific to this message:

- MessageContentCode must have a value of “3”
- InvTypeCode and RatePlanCode must be populated
- At least one of the following MUST be implemented:
  - LengthsOfStay
  - BookingLimit
  - RestrictionStatus

#### 3.1.7.1 Sample Scenario for Update Room/Rate Availability with LengthsOfStay, RestrictionStatus and BookingLimit

PMS “ABC” transmits the following ROOM/RATE availability update to CRS “123,” updating availability data for the period from January 1, 2013 through January 7, 2013.

- Hotel Code: HXCAIZZ
- Change applies to DOW: All Days
- Rate Plan: CR
- Room Type : A1K
- Number allotted: 5
- Status: Open
- Min LOS: 2

#### 3.1.7.1.1 Sample Request

```
<OTA_HotelAvailNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-08-01T09:30:47-05:00" EchoToken="89970240-068e-438b-ba11-1da35da19144" Target="Production" Version="1.001" TransactionIdentifier="938386" MessageContentCode="3">
  <!-- MessageContentCode=3 for Room/Rate Availability-->
  <AvailStatusMessages HotelCode="HXCAIZZ">
    <AvailStatusMessage BookingLimit="5" BookingLimitMessageType="SetLimit">
      <StatusApplicationControl Start="2013-01-01" End="2013-01-07"
    </StatusApplicationControl>
  </AvailStatusMessage>
  <AvailStatusMessage BookingLimit="5" BookingLimitMessageType="SetLimit">
    <StatusApplicationControl Start="2013-01-01" End="2013-01-07"
  </StatusApplicationControl>
  <InvTypeCode="A1K" RatePlanCode="CR"/>
    <LengthsOfStay>
      <LengthOfStay MinMaxMessageType="SetMinLOS" Time="2" TimeUnit="Day"/>
    </LengthsOfStay>
    <UniqueID Type="16" ID="1"/>
    <RestrictionStatus Restriction="Master" Status="Open"/>
  </AvailStatusMessage>
</AvailStatusMessages>
</OTA_HotelAvailNotifRQ>
```

#### 3.1.7.1.2 Sample Response

```
<OTA_HotelAvailNotifRS xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-08-01T09:30:47-05:00" EchoToken="89970240-068e-438b-ba11-1da35da19144" Target="Production" Version="1.001" TransactionIdentifier="938386">
  <Errors>
    <Error Type="13" Code="436" RecordID="1">Rate Plan Code CR does not exist</Error>
  </Errors>
</OTA_HotelAvailNotifRS>
```

### 3.1.8 Update Segment Availability

Defines the process between publisher and subscriber when a change to segment availability has occurred and an updated status must be communicated to the subscriber. The message assumes a push model, with the publishing system pushing the data to the subscriber.

Specific to this message:

- MessageContentCode must have a value of "4"
- RatePlanCategory must be populated
- At least one of the following MUST be implemented:
  - LengthsOfStay
  - BookingLimit
  - RestrictionStatus

### **3.1.8.1 Sample Scenario for Update Segment Availability with only RestrictionStatus**

PMS "ABC" transmits the following Segment/Category availability update to CRS "123," requesting availability data for the period from January 1, 2013 through January 7, 2013.

- Hotel Code: HXCAIZZ
- Change applies to DOW: SUN-MON
- Rate segment: P
- Status: Closed to Arrival

#### **3.1.8.1.1 Sample Request**

```
<OTA_HotelAvailNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-08-01T09:30:47-05:00" EchoToken="6d0a976b-60d1-45f3-9f95-d1e98c227884" Target="Production" Version="1.001" TransactionIdentifier="938385" MessageContentCode="4">
  <!-- MessageContentCode=4 for Segment Availability-->
  <AvailStatusMessages HotelCode="HXCAIZZ">
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-01-01" End="2013-01-07"
      RatePlanCategory="P" Mon="1" Tue="0" Weds="0" Thur="0" Fri="0" Sat="0" Sun="1"/>
      <UniqueID Type="16" ID="1"/>
      <RestrictionStatus Restriction="Arrival" Status="Close"/>
    </AvailStatusMessage>
  </AvailStatusMessages>
</OTA_HotelAvailNotifRQ>
```

#### **3.1.8.1.2 Sample Response**

```
<OTA_HotelAvailNotifRS xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-08-01T09:30:47-05:00" EchoToken="6d0a976b-60d1-45f3-9f95-d1e98c227884" Target="Production" Version="1.001" TransactionIdentifier="938385">
  <Success/>
</OTA_HotelAvailNotifRS>
```

### **3.1.9 Update Segment/Room Availability**

Defines the process between publisher and subscriber when a change to segment/room availability has occurred and an updated status must be communicated to the subscriber. The message assumes a push model, with the publishing system pushing the data to the subscriber.

Specific to this message:

- MessageContentCode must have a value of "5"
- InvTypeCode must be populated
- RatePlanCategory must be populated
- At least one of the following MUST be implemented:
  - LengthsOfStay
  - BookingLimit
  - RestrictionStatus



### **3.1.9.1 Sample Scenario for Update Segment/Room Availability with only LengthsOfStay**

PMS "ABC" transmits the following Segment/Room availability update to CRS "123."

- Hotel Code: HXCAIZZ
- Rate Segment: C
- Room Type: A1K
- Date / FPLOS: Jan 1 2013 / YNNNNYY
- Date / FPLOS: Jan 2 2013 / YNNNNYY
- Date / FPLOS: Jan 3 2013 / NNNYYYY

#### **3.1.9.1.1 Sample Request**

```
<OTA_HotelAvailNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-08-01T09:30:47-05:00" EchoToken="1c7ca126-54f3-4ad7-84a1-74680275ab59" Target="Production" Version="1.001" TransactionIdentifier="938387" MessageContentCode="5">
  <!-- MessageContentCode=5 for Segment/Room Availability-->
  <AvailStatusMessages HotelCode="HXCAIZZ">
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-01-01" End="2013-01-01"
      InvTypeCode="A1K" RatePlanCategory="C"/>
      <LengthsOfStay FixedPatternLength="7">
        <LengthOfStay MinMaxMessageType="FullPatternLOS">
          <LOS_Pattern FullPatternLOS="YNNNNYY"/>
        </LengthOfStay>
      </LengthsOfStay>
      <UniqueID Type="16" ID="1"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-01-02" End="2013-01-02"
      InvTypeCode="A1K" RatePlanCategory="C"/>
      <LengthsOfStay FixedPatternLength="7">
        <LengthOfStay MinMaxMessageType="FullPatternLOS">
          <LOS_Pattern FullPatternLOS="YNNNNYY"/>
        </LengthOfStay>
      </LengthsOfStay>
      <UniqueID Type="16" ID="2"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-01-03" End="2013-01-03"
      InvTypeCode="A1K" RatePlanCategory="C"/>
      <LengthsOfStay FixedPatternLength="7">
        <LengthOfStay MinMaxMessageType="FullPatternLOS">
          <LOS_Pattern FullPatternLOS="NNNNYYY"/>
        </LengthOfStay>
      </LengthsOfStay>
      <UniqueID Type="16" ID="3"/>
    </AvailStatusMessage>
  </AvailStatusMessages>
</OTA_HotelAvailNotifRQ>
```

#### **3.1.9.1.2 Sample Response**

```
<OTA_HotelAvailNotifRS xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-08-01T09:30:47-05:00" EchoToken="1c7ca126-54f3-4ad7-84a1-74680275ab59" Target="Production" Version="1.001" TransactionIdentifier="938387">
  <Success/>
</OTA_HotelAvailNotifRS>
```

### **3.1.10 Update House Availability**

This process describes the interaction between systems where notification of changes to House (Overall Property Level) availability must be communicated.

Specific to this message:

- MessageContentCode must have a value of "6"
- At least one of the following MUST be implemented:
  - BookingLimit
  - RestrictionStatus
  - LengthsOfStay

#### **3.1.10.1 Sample Scenario for Update House Availability with only BookingLimit**

PMS "ABC" transmits the following HOUSE availability update to CRS "123."

- Hotel Code: HXCAIZZ
- Date / Authorized Capacity: Jan 1 2013 / 350
- Date / Authorized Capacity: Jan 2 2013 / 348
- Date / Authorized Capacity: Jan 3 2013 / 354

##### **3.1.10.1.1 Sample Request**

```
<OTA_HotelAvailNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-08-01T09:30:47-05:00" EchoToken="edf194ca-e621-4a95-82b4-97c0b1cb59b8" Target="Production" Version="1.001" TransactionIdentifier="938388" MessageContentCode="6">
  <!-- MessageContentCode=6 for House Availability-->
  <AvailStatusMessages HotelCode="HXCAIZZ">
    <AvailStatusMessage BookingLimit="350" BookingLimitMessageType="SetLimit">
      <StatusApplicationControl Start="2013-01-01" End="2013-01-01"/>
      <UniqueID Type="16" ID="1"/>
    </AvailStatusMessage>
    <AvailStatusMessage BookingLimit="348" BookingLimitMessageType="SetLimit">
      <StatusApplicationControl Start="2013-01-02" End="2013-01-02"/>
      <UniqueID Type="16" ID="2"/>
    </AvailStatusMessage>
    <AvailStatusMessage BookingLimit="354" BookingLimitMessageType="SetLimit">
      <StatusApplicationControl Start="2013-01-03" End="2013-01-03"/>
      <UniqueID Type="16" ID="3"/>
    </AvailStatusMessage>
  </AvailStatusMessages>
</OTA_HotelAvailNotifRQ>
```

##### **3.1.10.1.2 Sample Response**

```
<OTA_HotelAvailNotifRS xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-08-01T09:30:47-05:00" EchoToken="edf194ca-e621-4a95-82b4-97c0b1cb59b8" Target="Production" Version="1.001" TransactionIdentifier="938388">
  <Success/>
</OTA_HotelAvailNotifRS>
```

### **3.1.11 Update Hurdle Availability**

This process describes the interaction between systems where notification of changes to Hurdle Rates availability must be communicated.

Specific to this message:

- MessageContentCode must have a value of "7"
- HurdleRate@Amount must be populated
- LengthsOfStay is optional
  - If LengthsOfStay is not used, it is up to trading partners to decide whether the hurdle is based upon average nightly rate or per day of stay when determining whether a given rate clears the hurdle.

#### **3.1.11.1 Sample Scenario for Update Hurdle Availability**

RMS "DEF" transmits the following Hurdle rate update to CRS "123."

- Hotel Code: HXCAIZZ
- Date / Hurdle Rate: Jan 1 2013 / € 100.00
- Date / Hurdle Rate: Jan 2 2013 / € 123.00
- Date / Hurdle Rate: Jan 3 2013 / € 98.00

##### **3.1.11.1.1 Sample Request**

```
<OTA_HotelAvailNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-08-01T09:30:47-05:00" EchoToken="eda39cb7-b92d-4778-a00c-a55131843ce0" Target="Production" Version="1.001" TransactionIdentifier="938389" MessageContentCode="7">
  <!-- MessageContentCode=7 for Hurdle Rate Update -->
  <AvailStatusMessages HotelCode="HXCAIZZ">
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-01-01" End="2013-01-01"/>
      <HurdleRate Amount="100.00" CurrencyCode="EUR"/>
      <UniqueID Type="16" ID="1"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-01-02" End="2013-01-02"/>
      <HurdleRate Amount="123.00" CurrencyCode="EUR"/>
      <UniqueID Type="16" ID="2"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-01-03" End="2013-01-03"/>
      <HurdleRate Amount="98.00" CurrencyCode="EUR"/>
      <UniqueID Type="16" ID="3"/>
    </AvailStatusMessage>
  </AvailStatusMessages>
</OTA_HotelAvailNotifRQ>
```

##### **3.1.11.1.2 Sample Response**

```
<OTA_HotelAvailNotifRS xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-08-01T09:30:47-05:00" EchoToken="eda39cb7-b92d-4778-a00c-a55131843ce0" Target="Production" Version="1.001" TransactionIdentifier="938389">
  <Success/>
</OTA_HotelAvailNotifRS>
```

### 3.1.12 *Update Best Available Rates*

Defines the process between publisher and subscriber when a change to BAR (Best Available Rate) at a specific property has occurred and an updated status must be communicated to the subscriber. The message assumes a push model, with the publishing system pushing the data to to the subscriber. Assume the example hurdle rate update request message from RMS “DEF” to CRS “123” shown in the previous section was re-sent the next day, on January 2nd 2013, due to some internal malfunction where the RMS did not mark the update as processed. This resulted in the first message trying to update the hurdle rate for a past date, January 1st, 2013. Assume also that RMS “DEF” and CRS “123” have identified this error condition as a warning and agreed it should not stop processing but return a warning element so technical support at RMS “DEF” can look at the situation and ensure there are no major issues with the interface. In this case, the response message from CRS “123” to RMS “DEF” would look as shown below. Notice attribute RecordID in the warning element matches the value of UniqueID for the element causing the warning in the request message.

Specific to this message:

- MessageContentCode must have a value of “11”
- RatePlanCode must be populated

#### 3.1.12.1 *Sample Scenario for Update Best Available Rates*

RMS “DEF” needs to transmit the following BestAvailableRate pricing control by LengthOfStay to CRS “123” for guests looking to book reservations arriving on either 6, 7 or 8 June 2013.

##### 3.1.12.1.1 *Sample Request*

```
<OTA_HotelAvailNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05" TimeStamp="2013-06-05T06:39:09" EchoToken="5f52f96c-9bcd-4a27-94ff-9f7ac3c22ef7" Target="Production" Version="1.001" TransactionIdentifier="938389" MessageContentCode="11">
  <!-- MessageContentCode=11 for Best Available Rate Update -->
  <AvailStatusMessages HotelCode="1699-0001">
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-06-06" End="2013-06-06"/>
      <BestAvailableRates>
        <BestAvailableRate RatePlanCode="BAR100" LengthOfStayTime="1"/>
        <BestAvailableRate RatePlanCode="BAR120" LengthOfStayTime="2"/>
        <BestAvailableRate RatePlanCode="BAR140" LengthOfStayTime="3"/>
      </BestAvailableRates>
      <UniqueID Type="16" ID="1"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-06-07" End="2013-06-07"/>
      <BestAvailableRates>
        <BestAvailableRate RatePlanCode="BAR120" LengthOfStayTime="1"/>
        <BestAvailableRate RatePlanCode="BAR140" LengthOfStayTime="2"/>
        <BestAvailableRate RatePlanCode="BAR180" LengthOfStayTime="3"/>
      </BestAvailableRates>
      <UniqueID Type="16" ID="2"/>
    </AvailStatusMessage>
  </AvailStatusMessages>
</OTA_HotelAvailNotifRQ>
```

```
<StatusApplicationControl Start="2013-06-08" End="2013-06-08"/>
<BestAvailableRates>
  <BestAvailableRate RatePlanCode="BAR140" LengthOfStayTime="1"/>
  <BestAvailableRate RatePlanCode="BAR160" LengthOfStayTime="2"/>
  <BestAvailableRate RatePlanCode="BAR160" LengthOfStayTime="3"/>
</BestAvailableRates>
<UniqueID Type="16" ID="3"/>
</AvailStatusMessage>
</AvailStatusMessages>
</OTA_HotelAvailNotifRQ>
```

### 3.1.12.1.2 Sample Response

```
<OTA_HotelAvailNotifRS xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2013-06-05T06:39:09" EchoToken="5f52f96c-9bcd-4a27-94ff-9f7ac3c22ef7" Target="Production" Version="1.001" TransactionIdentifier="938389">
  <Success/>
</OTA_HotelAvailNotifRS>
```

## 3.2 Get Availability

### 3.2.1 Overview

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

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

The Hotel Get Availability message pair provides the ability for a booking source to request availability status from a specified hotel property(s). The request message allows the booking source to resynchronize availability due to a system outage, or if a hotel property is behind a restrictive firewall and it is necessary to request availability.

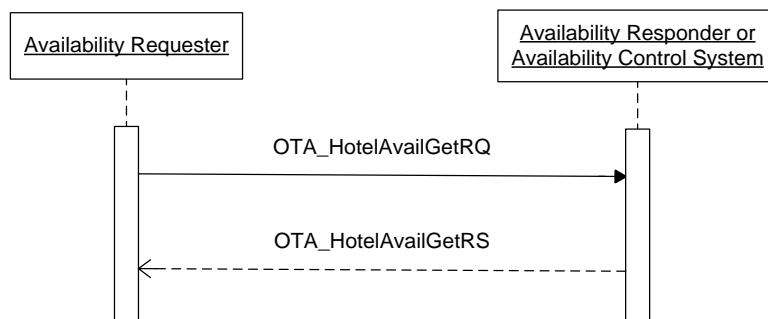
### 3.2.2 Roles

Role Name	Definition	Examples
Availability Requester	A system that has the need to obtain availability information.	<ul style="list-style-type: none"><li>Central Reservation System</li><li>Property Management System</li><li>Channel Management System</li><li>Revenue Management System</li></ul>
Availability Responder OR Availability Control System	A system that maintains availability information.	<ul style="list-style-type: none"><li>Central Reservation System</li><li>Property Management System</li><li>Channel Management System</li><li>Yield Management System</li></ul>

### 3.2.3 Use Case

Assumption:	Availability Requester and Availability Responder/Control System have agreed which attributes will be transmitted and accepted.
Pre-condition:	None
Trigger:	The Availability Requester, through event or manual action, has determined that it must obtain availability information from the Availability Responder/Control System.
Basic Course of Events:	<ol style="list-style-type: none"><li>1. Availability Requester constructs OTA_HotelAvailGetRQ with the appropriate query parameters.</li><li>2. Availability Requester successfully transmits message payload.</li><li>3. Availability Responder/Control System receives the message payload.</li><li>4. Availability Responder/Control System processes OTA_HotelAvailGetRQ.</li><li>5. Availability Responder/Control System returns the availability information appropriate to the query parameters, using OTA_HotelAvailGetRS.</li></ol>
Post-condition:	Availability Requester may update their database to log what was last successfully processed.
Exception Path:	None
Alternative Path:	None

### 3.2.4 Message Flows



### 3.2.5 Get Availability by Date Range

This process describes the interaction between systems where a list of available products can be obtained using a range of dates.

Specific to this message:

- HotelCode must be populated
- DateRange must be populated

#### 3.2.5.1 Sample Scenario

The CRS sends a request to hotel ABC123 requesting availability data for the period from March 1, 2013 through March 2, 2013.

The PMS responds with the list of availability by sellable products and their dates of availability containing full pattern length of stay restrictions.

- For the rate plan CORP and room type STD, only 2, 3 or 4 night reservations are allowed on March 1, 2013 while there is no availability for arrivals on March 2, 2013.

##### 3.2.5.1.1 Sample Request

```
<OTA_HotelAvailGetRQ xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-12-17T09:30:47-05:00" EchoToken="1d2cef5e-e1d3-4377-b4c0-bc0f96390931" Target="Production" Version="1.001">
  <POS>
    <Source>
      <RequestorID Type="22" ID="Passkey"/>
    </Source>
  </POS>
  <HotelAvailRequests>
    <HotelAvailRequest>
      <DateRange Start="2013-03-01" End="2013-03-02"/>
      <HotelRef ChainCode="YY" BrandCode="XX" HotelCode="ABC123"/>
    </HotelAvailRequest>
  </HotelAvailRequests>
</OTA_HotelAvailGetRQ>
```

##### 3.2.5.1.2 Sample Response

```
<OTA_HotelAvailGetRS xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-12-17T09:30:47-05:00" EchoToken="1d2cef5e-e1d3-4377-b4c0-bc0f96390931" Target="Production" Version="1.001">
  <Success/>
  <AvailStatusMessages ChainCode="YY" BrandCode="XX" HotelCode="ABC123">
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-03-01" End="2013-03-01"
RatePlanCodeType="RatePlanCode" RatePlanCode="CORP" InvCodeApplication="InvCode"
InvCode="STD" IsRoom="1" Override="1"/>
      <LengthsOfStay ArrivalDateBased="1" FixedPatternLength="8">
        <LengthOfStay Time="1" TimeUnit="Day" OpenStatusIndicator="0"
MinMaxMessageType="FullPatternLOS"/>
        <LengthOfStay Time="2" TimeUnit="Day" OpenStatusIndicator="1"
MinMaxMessageType="FullPatternLOS"/>
        <LengthOfStay Time="3" TimeUnit="Day" OpenStatusIndicator="1"
MinMaxMessageType="FullPatternLOS"/>
        <LengthOfStay Time="4" TimeUnit="Day" OpenStatusIndicator="1"
MinMaxMessageType="FullPatternLOS"/>
        <LengthOfStay Time="5" TimeUnit="Day" OpenStatusIndicator="0"
MinMaxMessageType="FullPatternLOS"/>
      </LengthsOfStay>
    </AvailStatusMessage>
  </AvailStatusMessages>
</OTA_HotelAvailGetRS>
```

```
<LengthOfStay Time="6" TimeUnit="Day" OpenStatusIndicator="0"
MinMaxMessageType="FullPatternLOS"/>
<LengthOfStay Time="7" TimeUnit="Day" OpenStatusIndicator="0"
MinMaxMessageType="FullPatternLOS"/>
<LengthOfStay Time="8" TimeUnit="Day" OpenStatusIndicator="0"
MinMaxMessageType="FullPatternLOS"/>
</LengthsOfStay>
</AvailStatusMessage>
<AvailStatusMessage>
<StatusApplicationControl Start="2013-03-02" End="2013-03-02"
RatePlanCodeType="RatePlanCode" RatePlanCode="CORP" InvCodeApplication="InvCode"
InvCode="STD" IsRoom="1" Override="1"/>
<LengthsOfStay ArrivalDateBased="1" FixedPatternLength="8">
<LengthOfStay Time="1" TimeUnit="Day" OpenStatusIndicator="0"
MinMaxMessageType="FullPatternLOS"/>
<LengthOfStay Time="2" TimeUnit="Day" OpenStatusIndicator="0"
MinMaxMessageType="FullPatternLOS"/>
<LengthOfStay Time="3" TimeUnit="Day" OpenStatusIndicator="0"
MinMaxMessageType="FullPatternLOS"/>
<LengthOfStay Time="4" TimeUnit="Day" OpenStatusIndicator="0"
MinMaxMessageType="FullPatternLOS"/>
<LengthOfStay Time="5" TimeUnit="Day" OpenStatusIndicator="0"
MinMaxMessageType="FullPatternLOS"/>
<LengthOfStay Time="6" TimeUnit="Day" OpenStatusIndicator="0"
MinMaxMessageType="FullPatternLOS"/>
<LengthOfStay Time="7" TimeUnit="Day" OpenStatusIndicator="0"
MinMaxMessageType="FullPatternLOS"/>
<LengthOfStay Time="8" TimeUnit="Day" OpenStatusIndicator="0"
MinMaxMessageType="FullPatternLOS"/>
</LengthsOfStay>
</AvailStatusMessage>
</AvailStatusMessages>
</OTA_HotelAvailGetRS>
```

### 3.2.6 Get Availability by Rate Plan

This process describes the interaction between systems where a list of available products can be obtained using a known rate plan.

Specific to this message:

- DateRange must be populated
- RatePlan must be populated

#### 3.2.6.1 Sample Scenario

The CRS system sends a request to hotel ABC123 requesting availability, in the form of booking limits and length of stay controls for the rates plan codes CORP and CORP2 from August 15, 2013 through October 1, 2013.

The PMS responds with availability for the rate plans of CORP and CORP2 and their restrictions from August 15, 2013 through October 1, 2013.

- The Booking Limit should be set on the rate plan of CORP. From August 15, 2013 through October 1, 2013, the limit should be set to 5.
- The Booking Limit should be set on the rate plan of CORP2. From August 15, 2013 through September 10, 2013, the limit should be set to 11 for all Mondays, Tuesdays,



Wednesdays, Thursdays and Fridays within the date range having the minimum length of stay of 2 days.

- From August 15, 2013 through September 10, 2013, Saturdays and Sundays have been closed and are not available for rate plan CORP2.
- The Booking Limit should be set on the rate plan of CORP2. From September 11, 2013 through October 1, 2013, the limit should be set to 5.

### 3.2.6.1.1 Sample Request

```
<OTA_HotelAvailGetRQ xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-12-17T09:30:47-05:00" EchoToken="56966596-eec3-48ac-8010-4f4ad681c357" Target="Production" Version="1.001">
  <POS>
    <Source>
      <RequestorID Type="22" ID="Passkey"/>
    </Source>
  </POS>
  <HotelAvailRequests>
    <HotelAvailRequest>
      <DateRange Start="2013-08-15" End="2013-10-01"/>
      <RatePlanCandidates>
        <RatePlanCandidate RatePlanCode="CORP"/>
        <RatePlanCandidate RatePlanCode="CORP2"/>
      </RatePlanCandidates>
      <LengthsOfStayCandidates SendLengthsOfStay="All"/>
      <HotelRef ChainCode="YY" BrandCode="XX" HotelCode="ABC123"/>
    </HotelAvailRequest>
  </HotelAvailRequests>
</OTA_HotelAvailGetRQ>
```

### 3.2.6.1.2 Sample Response

```
<OTA_HotelAvailGetRS xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-12-17T09:30:47-05:00" EchoToken="56966596-eec3-48ac-8010-4f4ad681c357" Target="Production" Version="1.001">
  <Success/>
  <AvailStatusMessages ChainCode="YY" BrandCode="XX" HotelCode="ABC123">
    <AvailStatusMessage BookingLimit="5" BookingLimitMessageType="SetLimit">
      <StatusApplicationControl Start="2013-08-15" End="2013-10-01"
RatePlanCode="CORP" />
      <UniqueID Type="16" ID="1" />
    </AvailStatusMessage>
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-08-15" End="2013-09-10"
RatePlanCode="CORP2" Mon="1" Tue="1" Weds="1" Thur="1" Fri="1" Sat="0" Sun="0"/>
      <LengthsOfStay>
        <LengthOfStay MinMaxMessageType="SetMinLOS" Time="2" TimeUnit="Day"/>
      </LengthsOfStay>
      <UniqueID Type="16" ID="2"/>
      <RestrictionStatus Status="Open"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-08-15" End="2013-09-10"
RatePlanCode="CORP2" Mon="0" Tue="0" Weds="0" Thur="0" Fri="0" Sat="1" Sun="1"/>
      <UniqueID Type="16" ID="3"/>
      <RestrictionStatus Status="Closed"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-09-11" End="2013-10-01"
RatePlanCode="CORP2"/>
      <UniqueID Type="16" ID="4" />
      <RestrictionStatus Status="Open"/>
    </AvailStatusMessage>
  </AvailStatusMessages>
</OTA_HotelAvailGetRS>
```

### 3.2.7 Get Availability by Room Type

This process describes the interaction between systems where a list of available products can be obtained using a known room type.

Specific to this message:

- DateRange must be populated
- RoomTypeCode must be populated

#### 3.2.7.1 Sample Scenario

The CRS system sends a request to hotel ABC123 requesting availability for the room type codes STD and DLX from June 11, 2013 through August 3, 2013. The Hotel Code is ABC123 and Room Types of STD & DLX.

The PMS responds with availability for the room types of STD and DLX and their restrictions from June 11, 2013 through August 3, 2013.

- The Booking Limit should be set on the room type of STD. From June 11, 2013 through August 3, 2013, the limit should be set to 55.
- The Booking Limit should be set on the rate plan of DLX. From June 11, 2013 through August 3, 2013, the limit should be set to 47 for all Fridays, Saturdays and Sundays within the date range having the minimum length of stay of 3 days.

##### 3.2.7.1.1 Sample Request

```
<OTA_HotelAvailGetRQ xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-12-17T09:30:47-05:00" EchoToken="84923a59-916c-434d-8f18-0887dbd03af5"
Target="Production" Version="1.001">
  <POS>
    <Source >
      <RequestorID Type="22" ID="Passkey"/>
    </Source>
  </POS>
  <HotelAvailRequests>
    <HotelAvailRequest SendBookingLimit="1" BookingLimitMessageType="SetLimit">
      <DateRange Start="2013-06-11" End="2013-08-03"/>
      <HotelRef ChainCode="YY" BrandCode="XX" HotelCode="ABC123"/>
      <RoomTypeCandidates>
        <RoomTypeCandidate RoomTypeCode="STD" />
        <RoomTypeCandidate RoomTypeCode="DLX" />
      </RoomTypeCandidates>
      <LengthsOfStayCandidates SendLengthsOfStay="All"/>
    </HotelAvailRequest>
  </HotelAvailRequests>
</OTA_HotelAvailGetRQ>
```

##### 3.2.7.1.2 Sample Response

```
<OTA_HotelAvailGetRS xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-12-17T09:30:47-05:00" EchoToken="84923a59-916c-434d-8f18-0887dbd03af5"
Target="Production" Version="1.001">
  <Success/>
  <AvailStatusMessages ChainCode="YY" BrandCode="XX" HotelCode="ABC123" >
```

```
<AvailStatusMessage BookingLimit="55" BookingLimitMessageType="SetLimit">
  <StatusApplicationControl Start="2013-06-11" End="2013-08-03" InvCode="STD" />
  <UniqueID Type="16" ID="1"/>
</AvailStatusMessage>
<AvailStatusMessage BookingLimit="47" BookingLimitMessageType="SetLimit">
  <StatusApplicationControl Start="2013-06-11" End="2013-08-03" InvCode="DLX"
Mon="0" Tue="0" Weds="0" Thur="0" Fri="1" Sat="1" Sun="1"/>
  <LengthsOfStay>
    <LengthOfStay MinMaxMessageType="SetMinLOS" Time="3" TimeUnit="Day"/>
  </LengthsOfStay>
  <UniqueID Type="16" ID="2"/>
</AvailStatusMessage>
</AvailStatusMessages>
</OTA_HotelAvailGetRS>
```

### 3.2.8 Get Availability by Rate Plan/Room Type Combination

This process describes the interaction between systems where a list of available products can be obtained using a combination of rate plan and room type.

Specific to this message:

- DateRange must be populated
- RoomType must be populated
- RatePlan must be populated

#### 3.2.8.1 Sample Scenario

The CRS system sends a request to hotel ABC123 requesting availability for the room type code and rate plan code combination of STD and CORP, from June 11, 2013 through August 3, 2013. The Hotel Code is ABC123, Rate Plan is CORP and Room Type of STD.

The PMS responds with availability for the combination of the room type of STD and the rate plan of CORP and their restrictions from June 11, 2013 through August 3, 2013.

- The Booking Limit should be set on the combination of the room type of STD and rate plan of CORP. From June 11, 2013 through August 3, 2013, the limit should be set to 35 for all Mondays, Tuesdays, Wednesdays, Thursdays and Fridays within the date range having the minimum length of stay of 2 days.

##### 3.2.8.1.1 Sample Request

```
<OTA_HotelAvailGetRQ xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-12-17T09:30:47-05:00" EchoToken="15f6cf5c-6b9e-4939-b98e-b24fd6ea8eb5"
Target="Production" Version="1.001">
  <POS>
    <Source>
      <RequestorID Type="22" ID="Passkey" />
    </Source>
  </POS>
  <HotelAvailRequests>
    <HotelAvailRequest>
      <DateRange Start="2013-06-11" End="2013-08-03"/>
      <RatePlanCandidates>
        <RatePlanCandidate RatePlanCode="CORP" />
      </RatePlanCandidates>
      <RoomTypeCandidates>
        <RoomTypeCandidate RoomTypeCode="STD" />
      </RoomTypeCandidates>
    </HotelAvailRequest>
  </HotelAvailRequests>
</OTA_HotelAvailGetRQ>
```

```
</RoomTypeCandidate>
</RoomTypeCandidates>
<LengthsOfStayCandidates SendLengthsOfStay="All"/>
<HotelRef ChainCode="YY" BrandCode="XX" HotelCode="ABC123"/>
</HotelAvailRequest>
</HotelAvailRequests>
</OTA_HotelAvailGetRQ>
```

### 3.2.8.1.2 Sample Response

```
<OTA_HotelAvailGetRS xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2012-12-17T09:30:47-05:00" EchoToken="15f6cf5c-6b9e-4939-b98e-b24fd6ea8eb5"
Target="Production" Version="1.001">
  <Success/>
  <AvailStatusMessages ChainCode="YY" BrandCode="XX" HotelCode="ABC123" >
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-06-11" End="2013-08-03"
RatePlanCode="CORP" InvCode="STD" Mon="1" Tue="1" Weds="1" Thur="1" Fri="1" Sat="0"
Sun="0"/>
      <LengthsOfStay>
        <LengthOfStay MinMaxMessageType="SetMinLOS" Time="2" TimeUnit="Day"/>
      </LengthsOfStay>
      <UniqueID Type="16" ID="1"/>
      <RestrictionStatus Status="Open"/>
    </AvailStatusMessage>
  </AvailStatusMessages>
</OTA_HotelAvailGetRS>
```

## 3.2.9 Get Best Available Rates

This process describes the interaction between systems where a list of available products can be obtained using a range of dates.

Specific to this message:

- DateRange must be populated
- SendRatePlanCode must be populated

### 3.2.9.1 Sample Scenario

The CRS system sends a request to the RMS system for hotel ABC123 requesting best available rate controls, by length of stay, from June 20, 2013 through June 22, 2013. The request asks for the rate plan codes in the response. The RMS responds with best available rate controls for the date range giving a best available rate per arrival day per length of stay.

Note: If the rate plan amount is needed, use the OTA\_HotelRatePlanRQ message in the HTNG Product Distribution Rates Specification.

#### 3.2.9.1.1 Sample Request

```
<OTA_HotelAvailGetRQ xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2013-01-21T09:30:47-06:00" EchoToken="5ad42eb1-94e7-4370-a82f-f2b647c748ff"
Target="Production" Version="1.001">
  <HotelAvailRequests>
    <HotelAvailRequest>
      <DateRange Start="2013-06-20" End="2013-06-22"/>
      <BestAvailableRateCandidate SendLengthOfStayTime="1" SendRatePlanCode="1"/>
    </HotelAvailRequest>
  </HotelAvailRequests>
</OTA_HotelAvailGetRQ>
```

```

    <HotelRef ChainCode="YY" BrandCode="XX" HotelCode="ABC123"/>
  </HotelAvailRequest>
</HotelAvailRequests>
</OTA_HotelAvailGetRQ>

```

### 3.2.9.1.2 Sample Response

```

<OTA_HotelAvailGetRS xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2013-01-
21T09:30:47-06:00" EchoToken="5ad42eb1-94e7-4370-a82f-f2b647c748ff"
Target="Production" Version="1.001">
  <Success/>
  <AvailStatusMessages ChainCode="YY" BrandCode="XX" HotelCode="ABC123">
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-06-20" End="2013-06-20"/>
      <BestAvailableRates>
        <BestAvailableRate RatePlanCode="BAR3" LengthOfStayTime="1"/>
        <BestAvailableRate RatePlanCode="BAR3" LengthOfStayTime="2"/>
        <BestAvailableRate RatePlanCode="BAR3" LengthOfStayTime="3"/>
        <BestAvailableRate RatePlanCode="BAR3" LengthOfStayTime="4"/>
        <BestAvailableRate RatePlanCode="BAR3" LengthOfStayTime="5"/>
        <BestAvailableRate RatePlanCode="BAR3" LengthOfStayTime="6"/>
        <BestAvailableRate RatePlanCode="BAR3" LengthOfStayTime="7"/>
        <BestAvailableRate RatePlanCode="BAR3" LengthOfStayTime="8"/>
      </BestAvailableRates>
      <UniqueID Type="16" ID="1"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-06-21" End="2013-06-21"/>
      <BestAvailableRates>
        <BestAvailableRate RatePlanCode="BAR4" LengthOfStayTime="1"/>
        <BestAvailableRate RatePlanCode="BAR4" LengthOfStayTime="2"/>
        <BestAvailableRate RatePlanCode="BAR4" LengthOfStayTime="3"/>
        <BestAvailableRate RatePlanCode="BAR4" LengthOfStayTime="4"/>
        <BestAvailableRate RatePlanCode="BAR4" LengthOfStayTime="5"/>
        <BestAvailableRate RatePlanCode="BAR4" LengthOfStayTime="6"/>
        <BestAvailableRate RatePlanCode="BAR4" LengthOfStayTime="7"/>
        <BestAvailableRate RatePlanCode="BAR4" LengthOfStayTime="8"/>
      </BestAvailableRates>
      <UniqueID Type="16" ID="2"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-06-22" End="2013-06-22"/>
      <BestAvailableRates>
        <BestAvailableRate RatePlanCode="BAR3" LengthOfStayTime="1"/>
        <BestAvailableRate RatePlanCode="BAR2" LengthOfStayTime="2"/>
        <BestAvailableRate RatePlanCode="BAR2" LengthOfStayTime="3"/>
        <BestAvailableRate RatePlanCode="BAR2" LengthOfStayTime="4"/>
        <BestAvailableRate RatePlanCode="BAR2" LengthOfStayTime="5"/>
        <BestAvailableRate RatePlanCode="BAR2" LengthOfStayTime="6"/>
        <BestAvailableRate RatePlanCode="BAR4" LengthOfStayTime="7"/>
        <BestAvailableRate RatePlanCode="BAR4" LengthOfStayTime="8"/>
      </BestAvailableRates>
      <UniqueID Type="16" ID="3"/>
    </AvailStatusMessage>
  </AvailStatusMessages>
</OTA_HotelAvailGetRS>

```

### 3.2.10 Get Rate Hurdles

This process describes the interaction between systems where a list of hurdle amounts can be obtained using a range of dates.

Specific to this message:

- DateRange must be populated
- SendAmount must be populated

### **3.2.10.1     *Sample Scenario***

The CRS system sends a request to the RMS system for hotel ABC123 requesting hurdle rate controls, including delta values, from June 20, 2013 through June 22, 2013.

The RMS responds with best available rate controls for the date range giving a best available rate per stay day.

#### **3.2.10.1.1     *Sample Request***

```
<OTA_HotelAvailGetRQ xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2013-01-21T09:30:47-06:00" EchoToken="bebd86d3-6ee9-4bd9-a298-7367eb26bcc0"
Target="Production" Version="1.001">
  <HotelAvailRequests>
    <HotelAvailRequest>
      <DateRange Start="2013-06-20" End="2013-06-22"/>
      <HurdleRateCandidate SendAmount="1"/>
      <DeltaCandidate SendAmount="1"/>
      <HotelRef ChainCode="YY" BrandCode="XX" HotelCode="ABC123"/>
    </HotelAvailRequest>
  </HotelAvailRequests>
</OTA_HotelAvailGetRQ>
```

#### **3.2.10.1.2     *Sample Response***

```
<OTA_HotelAvailGetRS xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2013-01-21T09:30:47-06:00" EchoToken="bebd86d3-6ee9-4bd9-a298-7367eb26bcc0"
Target="Production" Version="1.001">
  <Success/>
  <AvailStatusMessages ChainCode="YY" BrandCode="XX" HotelCode="ABC123">
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-06-20" End="2013-06-20"/>
      <HurdleRate Amount="107.50"/>
      <Delta Amount="2.25" Ceiling="35" MaxSold="123"/>
      <UniqueID Type="16" ID="1"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-06-21" End="2013-06-21"/>
      <HurdleRate Amount="99.80"/>
      <Delta Amount="12.00" Ceiling="17" MaxSold="88"/>
      <UniqueID Type="16" ID="2"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-06-22" End="2013-06-22"/>
      <HurdleRate Amount="111.00"/>
      <Delta Amount="0.75" Ceiling="103" MaxSold="105"/>
      <UniqueID Type="16" ID="3"/>
    </AvailStatusMessage>
  </AvailStatusMessages>
</OTA_HotelAvailGetRS>
```

### **3.2.11     *Get Availability Restrictions***

This process describes the interaction between systems where a list of available products can be obtained using a range of dates.

Specific to this message:

- DateRange must be populated
- Desired Status must be populated

### 3.2.11.1 Sample Scenario

The CRS sends a request to the RMS for hotel ABC123 requesting only "Open" and "Close" restrictions from June 20, 2013 through June 22, 2013. The PMS responds with the following restrictions from June 20, 2013 through June 22, 2013:

Start Date	End Date	Hotel Code	Rate Plan	Room Type	Status	Type
6/20/2013	6/20/2013	ABC123	All	All	Close	Master
6/21/2013	6/22/2013	ABC123	All	All	Open	Master
6/20/2013	6/22/2013	ABC123	RACK	GR	Close	Arrival

#### 3.2.11.1.1 Sample Request

```
<OTA_HotelAvailGetRQ xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2013-01-21T09:30:47-06:00" EchoToken="962bcfb1-03f9-4bbe-9b94-4173f3d3f975"
Target="Production" Version="1.001">
  <HotelAvailRequests>
    <HotelAvailRequest>
      <DateRange Start="2013-06-20" End="2013-06-22"/>
      <RestrictionStatusCandidates SendAllRestrictions="0">
        <RestrictionStatusCandidate Status="Close" />
        <RestrictionStatusCandidate Status="Open" />
      </RestrictionStatusCandidates>
      <HotelRef ChainCode="YY" BrandCode="XX" HotelCode="ABC123"/>
    </HotelAvailRequest>
  </HotelAvailRequests>
</OTA_HotelAvailGetRQ>
```

#### 3.2.11.1.2 Sample Response

```
<OTA_HotelAvailGetRS xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2013-01-21T09:30:47-06:00" EchoToken="962bcfb1-03f9-4bbe-9b94-4173f3d3f975"
Target="Production" Version="1.001">
  <Success/>
  <AvailStatusMessages ChainCode="YY" BrandCode="XX" HotelCode="ABC123">
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-06-20" End="2013-06-20"/>
      <RestrictionStatus Status="Close" Restriction="Master"/>
      <UniqueID Type="16" ID="1"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-06-21" End="2013-06-22"/>
      <RestrictionStatus Status="Open" Restriction="Master"/>
      <UniqueID Type="16" ID="2"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-06-20" End="2013-06-22"
RatePlanCode="RACK" InvTypeCode="GR"/>
      <RestrictionStatus Status="Close" Restriction="Arrival"/>
      <UniqueID Type="16" ID="2"/>
    </AvailStatusMessage>
  </AvailStatusMessages>
</OTA_HotelAvailGetRS>
```

### 3.2.12 *Get Length of Stay Restrictions*

This process describes the interaction between systems where a list of available products can be obtained using a range of dates.

Specific to this message:

- DateRange must be populated
- SendAllLengthOfStay must contain a value of "1" or "true"

#### 3.2.12.1 *Sample Scenario*

The CRS sends a request to the RMS system for hotel ABC123 requesting all Lengths of Stay from June 20, 2013 through June 22, 2013.

The PMS responds with the following lengths of stay from June 20, 2013 through June 22, 2013:

Start Date	End Date	Hotel Code	RatePlan	Room Type	Restriction	Length (LOS)
6/20/2013	6/20/2013	ABC123	All	All	MinLOS	2
6/20/2013	6/20/2013	ABC123	All	All	MaxLOS	5
6/21/2013	6/22/2013	ABC123	All	All	MinLOS	1
6/20/2013	6/22/2013	ABC123	RACK	KS	MinLOS	3
6/20/2013	6/22/2013	ABC123	RACK	KS	MaxLOS	9

##### 3.2.12.1.1 *Sample Request*

```
<OTA_HotelAvailGetRQ xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2013-01-21T09:30:47-06:00" EchoToken="1f6b60c8-81e5-4560-90aa-8ecbcb497aea"
Target="Production" Version="1.001">
  <HotelAvailRequests>
    <HotelAvailRequest>
      <DateRange Start="2013-06-20" End="2013-06-22"/>
      <LengthsOfStayCandidates SendAllLengthsOfStay="1"/>
      <HotelRef ChainCode="YY" BrandCode="XX" HotelCode="ABC123"/>
    </HotelAvailRequest>
  </HotelAvailRequests>
</OTA_HotelAvailGetRQ>
```

##### 3.2.12.1.2 *Sample Response*

```
<OTA_HotelAvailGetRS xmlns="http://www.opentravel.org/OTA/2003/05" Timestamp="2013-01-21T09:30:47-06:00" EchoToken="1f6b60c8-81e5-4560-90aa-8ecbcb497aea"
Target="Production" Version="1.001">
  <Success/>
  <AvailStatusMessages ChainCode="YY" BrandCode="XX" HotelCode="ABC123">
    <AvailStatusMessage>
      <StatusApplicationControl Start="2013-06-20" End="2013-06-20"/>
      <LengthsOfStay ArrivalDateBased="1">
        <LengthOfStay Time="2" TimeUnit="Day" MinMaxMessageType="MinLOS"/>
        <LengthOfStay Time="5" TimeUnit="Day" MinMaxMessageType="MaxLOS"/>
      </LengthsOfStay>
    </AvailStatusMessage>
  </AvailStatusMessages>
</OTA_HotelAvailGetRS>
```



```

        </LengthOfStay>
        <UniqueID Type="16" ID="1"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
        <StatusApplicationControl Start="2013-06-21" End="2013-06-22"/>
        <LengthOfStay Time="1" TimeUnit="Day" MinMaxMessageType="MinLOS"/>
        <UniqueID Type="16" ID="2"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
        <StatusApplicationControl RatePlanCode="RACK" InvCode="KS" Start="2013-06-20"
End="2013-06-22" />
        <LengthOfStay Time="3" TimeUnit="Day" MinMaxMessageType="MinLOS"/>
        <LengthOfStay Time="9" TimeUnit="Day" MinMaxMessageType="MaxLOS"/>
        <UniqueID Type="16" ID="3"/>
    </AvailStatusMessage>
</AvailStatusMessages>
</OTA_HotelAvailGetRS>

```

## 3.3 Update Inventory

### 3.3.1 Overview

This process describes the interaction between systems where notification of changes to inventory must be communicated.

The Hotel Inventory Count message pair provides the ability for a Reservation Broker to request inventory amounts from a Reservation Server. The request message allows the Reservation Broker to resynchronize inventory due to a system outage or if a Reservation Service is behind a restrictive firewall, and it is necessary to request inventory.

### 3.3.2 Roles

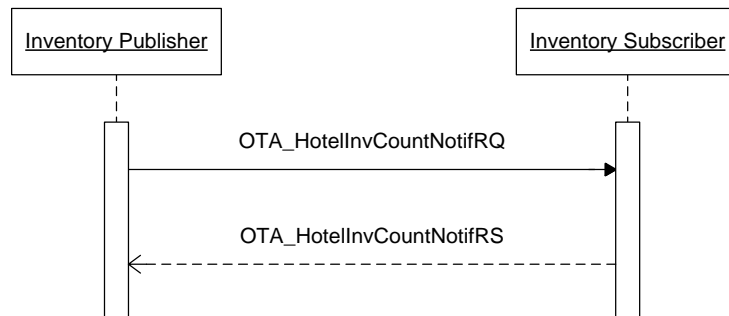
Role Name	Definition	Examples
Inventory Publisher	A system that maintains inventory and notifies Inventory Subscriber(s) of changes.	<ul style="list-style-type: none"> <li>Central Reservation System</li> <li>Property Management System</li> <li>Channel Management System</li> </ul>
Inventory Subscriber	A system that requires notification of inventory changes as they occur in Inventory Publisher system.	<ul style="list-style-type: none"> <li>Central Reservation System</li> <li>Property Management System</li> <li>Channel Management System</li> <li>Revenue Management System</li> <li>Online Travel Agent</li> </ul>

### 3.3.3 Use Case

Assumptions:	<ul style="list-style-type: none"> <li>Publisher and Subscriber have agreed which attributes will be transmitted and accepted.</li> <li>Publisher and Subscriber have agreed upon a notification model (either real-time or timed interval).</li> </ul>
Pre-condition:	None

Trigger:	A change to inventory occurs in the Inventory Publisher's system.
Basic Course of Events:	<ol style="list-style-type: none"><li>1. Inventory Publisher constructs OTA_HotelInvCountNotifRQ containing the availability change(s).</li><li>2. Inventory Publisher successfully transmits message payload.</li><li>3. Inventory Subscriber receives the message payload.</li><li>4. Inventory Subscriber processes OTA_HotelInvCountNotifRQ.</li><li>5. Inventory Subscriber acknowledges with the OTA_HotelInvCountNotifRS indicating processing status of the message.</li><li>6. Inventory Publisher may update their database to log what was last successfully processed by the Subscriber.</li></ol>
Post-condition:	Inventory Subscriber may update their log with what was last successfully processed.
Exception Path 1:	<ol style="list-style-type: none"><li>1. At step 3, the Inventory Subscriber does not successfully receive the message.</li><li>2. Inventory Subscriber does not return the OTA_HotelInvCountNotifRS.</li><li>3. Inventory Publisher system may:<ol style="list-style-type: none"><li>a. Log the unsuccessful send.</li><li>b. Resend the update notification immediately.</li><li>c. Re-queue the update notification.</li><li>d. Wait until the next agreed interval to send an update notification.</li></ol></li></ol>
Exception Path 2:	<ol style="list-style-type: none"><li>1. At Step 4, the Inventory Subscriber does not successfully process the message.</li><li>2. Inventory Subscriber returns OTA_HotelInvCountNotifRS with the appropriate error indicator.</li><li>3. Inventory Publisher system may:<ol style="list-style-type: none"><li>a. Log the response error.</li><li>b. Take action to correct the error and/or resend the transaction if necessary.</li><li>c. Re-queue the update notification.</li><li>d. Wait until the next agreed interval to send an update notification.</li></ol></li></ol>
Alternative Path:	If the Subscriber wishes to take on role of <i>Inventory Requester</i> , it then may perform a <a href="#">Get Inventory</a> .

### 3.3.4 Message Flows



### 3.3.5 Update Hotel Inventory

This process describes the interaction between systems when inventory values are being updated for a hotel.

Specific to this message:

- AllInvCode

#### 3.3.5.1 Sample Scenario

##### 3.3.5.1.1 Sample Request

```
<OTA_HotelInvCountNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05"
TimeStamp="2013-10-01T09:41:51.982" EchoToken="25e86d37-bebd-4da4-8cd9-bfe13aa83b8c"
Target="Production" Version="1.001">
  <Inventories HotelCode="10001" HotelName="Test Hotel">
    <Inventory>
      <UniqueID Type="16" ID="1" />
      <StatusApplicationControl Start="2014-03-01" End="2014-03-01"
AllInvCode="true"/>
      <InvCounts>
        <InvCount CountType="1" Count="20" />
        <InvCount CountType="2" Count="17" />
        <InvCount CountType="6" Count="1" />
        <InvCount CountType="8" Count="2" />
      </InvCounts>
    </Inventory>
  </Inventories>
</OTA_HotelInvCountNotifRQ>
```

##### 3.3.5.1.2 Sample Response

```
<OTA_HotelInvCountNotifRS xmlns="http://www.opentravel.org/OTA/2003/05"
TimeStamp="2013-10-01T09:41:51.982" EchoToken="25e86d37-bebd-4da4-8cd9-bfe13aa83b8c"
Target="Production" Version="1.001">
  <Success/>
</OTA_HotelInvCountNotifRS>
```

### **3.3.6 Update Room Type Inventory**

This process describes the interaction between systems when inventory values are being updated for a Room Type.

Specific to this message:

- InvTypeCode

#### **3.3.6.1 Sample Scenario**

##### **3.3.6.1.1 Sample Request**

```
<OTA_HotelInvCountNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05"
TimeStamp="2013-10-01T09:41:51.982" EchoToken="48ab3f94-49e2-418a-97fb-2cec9c5937bd"
Target="Production" Version="1.001">
  <Inventories HotelCode="10001" HotelName="Test Hotel">
    <Inventory>
      <UniqueID Type="16" ID="1" />
      <StatusApplicationControl Start="2014-03-01" End="2014-03-01"
InvTypeCode="King"/>
      <InvCounts>
        <InvCount CountType="1" Count="20" />
        <InvCount CountType="2" Count="17" />
        <InvCount CountType="6" Count="1" />
        <InvCount CountType="8" Count="2" />
      </InvCounts>
    </Inventory>
  </Inventories>
</OTA_HotelInvCountNotifRQ>
```

##### **3.3.6.1.2 Sample Response**

```
<OTA_HotelInvCountNotifRS xmlns="http://www.opentravel.org/OTA/2003/05"
TimeStamp="2013-10-01T09:41:51.982" EchoToken="48ab3f94-49e2-418a-97fb-2cec9c5937bd"
Target="Production" Version="1.001">
  <Success/>
</OTA_HotelInvCountNotifRS>
```

## 4 Messages

Please note that conditionally mandatory items (outlined in the above use cases) are marked with an asterisk (\*).

### 4.1 Update Availability

#### 4.1.1 Data Element Table – Update Availability Request

Element   @Attribute	Num	Description/Contents
OTA_HotelAvailNotifRQ	1	Root element of the message.
@EchoToken	0..1	A reference for additional message identification, assigned by the requesting host system. When a request message includes an echo token the corresponding response message MUST include an echo token with an identical value.
@TimeStamp	1	Time of the transaction.
@Version	1	Version is a mandatory attribute in OTA; therefore, it must remain Mandatory in HTNG in order to be able to use the same message.
@TransactionIdentifier	0..1	If the availability update is the result of a reservation upload, the reservation number should be indicated in this field.
@MessageContentCode	1	The attribute refers to OpenTravel Alliance code list MCC which includes RateAvail, RoomAvail, RoomRateAvail, SegmentAvail, SegmentRoomAvail, HouseAvail and HurdleRateUpdate.
OTA_HotelAvailNotifRQ / AvailStatusMessages	1	Must be sent for the message to have a meaning.
@HotelCode	1	This is the code of the property whose availability is being updated.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage	1..n	Although all restrictions are optional, at least one should be sent for the message to have any meaning. Upper limit to be defined by trading partners.
@BookingLimit	0..1	This is the maximum number of rooms for the InvTypeCode sent that can be booked. Although all restrictions are Optional, at least one should be sent for the message to have any meaning.
@BookingLimitMessageType	0..1	Enumerated values are used to indicate whether the booking limit sent in the transmission is used to set, adjust or delete the booking limit (SetLimit, AdjustLimit, RemoveLimit).
@BookingThreshold	0..1	This is the number of rooms that can be sold down to. This value may be negative to allow for overbookings.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / UniqueID	1	The unique identifier element allows the trading partners to uniquely identify each AvailStatusMessage for tracing of transactions.

Element   @Attribute	Num	Description/Contents
@Type	1	Refers to OpenTravel Alliance code list UIT - nr 16 = Reference. This is used to identify each single availability status message for error reporting purposes.
@ID	1	A unique incremental number for each availability message that identifies that specific message.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / StatusApplicationControl	1	Element defining application of controls being sent.
@Start, @End	1	The first and last dates for which the availability update is being sent.
@Mon,@Tue,@Weds,@Thur,@Fri,@Sat,@Sun	0..1	The day of the week indicators are used to communicate which days of the week the update pertains to. If one is sent, they must all be sent.
@InvTypeCode	0..1*	This is the room type code for which the update is being sent.
@RatePlanCode	0..1*	This is the rate plan whose availability is being updated.
@RatePlanCategory	0..1*	This is the Category or segment to which the rate whose availability is being updated belongs.
@InvCodeApplication	0..1*	This clarifies whether the InvTypeCode is an actual room type code OR a room grouping (for instance, a room group such as "deluxe" which includes multiple room types). The only two enumerations allowed would be InvCode for room type and InvGroupingCode for a room group. Partners will need to agree upon implementation whether the room grouping is supported by both.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / HurdleRate	0..1*	Restriction based on the minimum rate to be considered for availability, ex., can sell weekend rate only if charging the hurdle rate or more.
@Amount	1	This is the Hurdle rate the user wishes to set for the season sent.
@CurrencyCode	0..1	Currency of delta adjustment using ISO 4217 codes. If the partner receiving the data does not support currency code for Hurdle rate, the currency code would be ignored.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / Delta	0..1	Incremental adjustment to the hurdle rate. Included if supported by trading partners.
@Amount	0..1	Amount of delta adjustment.

Element   @Attribute	Num	Description/Contents
@CurrencyCode	0..1	Currency of delta adjustment using ISO 4217 codes. If the partner receiving the data does not support currency code for Hurdle rate, the currency code would be ignored.
@Ceiling, @MaxSold	0..1	Limits on the application of the Delta.
@InvTypeCode	0..1	This is the Room Type for which the Hurdle rate is sent (if sent on a room type by room type basis).
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / RestrictionStatus	0..1*	If no BookingLimit or LengthOfStay restriction is sent, then the Status field must be sent for the message to be meaningful.
@Restriction	1	Enumeration; possible values = "Master," "Arrival," "Departure." Works in conjunction with @Status.  If one of the partners implementing the message does not support all the enumerations, the enumerations that are not supported should be mapped to supported enumerations by the receiving system. An unsupported enumeration would be ignored by the receiving system; therefore, the type of status messages must be agreed upon implementation between the two partners.
@Status	1	Enumeration; possible values = "Open," "Close," "OnRequest." Works in conjunction with @Restriction.  If one of the partners implementing the message does not support all the enumerations, the enumerations that are not supported should be mapped to supported enumerations by the receiving system. An unsupported enumeration would be ignored by the receiving system; therefore, the type of status messages must be agreed upon implementation between the two partners.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / BestAvailableRates	0..1*	A collection of the best rates available.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / BestAvailableRates / BestAvailableRate	1..n	This element is normally repeated for each length of stay. The maximum length of stay "n" is to be agreed upon by trading partners, though normally it is 8.  In cases where a single BAR value is applicable for all length of stays, the element is sent once.
@RatePlanCode	1	This is the code identifying the best available rate.

Element   @Attribute	Num	Description/Contents
@LengthOfStayTime	1	Length of stay. This is normally a number from 1 to “n” (defined above).  In cases where a single BAR value is applicable for all length of stays, the value is zero.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay	0..1*	If no BookingLimit or Status restriction is sent, then the LengthsOfStay field must be sent with some LOS restrictions in order for the message to be meaningful.
@ArrivalDateBased	0..1	True indicates that LOS is based on arrival date. False indicates that LOS is based on stay date. If not present, this value is assumed to be true.
@FixedPatternLength	0..1	If both partners agree to send the Full Pattern LOS then this field is set to the length of the pattern string further inside the message. For example, this would be set to 7 when the string has seven characters, from LOS1 to LOS7+.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay / LengthOfStay	1..n	Conditionally Mandatory. If LengthsOfStay is sent then there must be at least one LengthOfStay element.
@MinMaxMessageType	0..1	MinMaxLengthOfStay set = SetMinLOS when sending MinLOS, the default setting. It can be set = FullPatternLOS if both partners support the Full Pattern Length of Stay as available in the OpenTravel Alliance message. Other values could be used based on partner agreement.
@Time, @TimeUnit	0..1	If sending MinLOS then TimeUnit set = Day and Time set to the MinLOS value.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay / LengthOfStay / LOS_Pattern	0..1	Used to define the LOS Pattern.
@FullPatternLOS	0..1	This is set to the string with ‘Y’ for the LOS that are open and ‘N’ for the LOS that are closed. The length of the string is indicated in @FixedPatternLength described above. For example, to open the odd LOS (LOS1, LOS3, LOS5 and LOS7) and close the even LOS (LOS2, LOS4 and LOS6) then the string would be “YNYNYNY.”

#### 4.1.2 Data Element Table – Update Availability Response

Element   @Attribute	Num	Description/Contents
OTA_HotelAvailNotifRS	1	Root element of the message.



Element   @Attribute	Num	Description/Contents
@EchoToken	0..1	A reference for additional message identification, assigned by the requesting host system. When a request message includes an echo token the corresponding response message MUST include an echo token with an identical value.
@Version	1	Version is a mandatory attribute in OTA; therefore, it must remain Mandatory in HTNG in order to be able to use the same message.
@TimeStamp	1	Time of the transaction.
@MessageContentCode	1	The attribute refers to OpenTravel Alliance code list MCC which includes RateAvail, RoomAvail, RoomRateAvail, SegmentAvail, SegmentRoomAvail, HouseAvail and HurdleRateUpdate.
OTA_HotelAvailNotifRS / Success	0..1	This is the annotation that the availability status message batch was received successfully. It could be combined with warning messages if some of the messages in the batch had issues.
OTA_HotelAvailNotifRS / Warnings	0..1	Used when a message has been successfully processed to report any warnings or business errors that occur.
OTA_HotelAvailNotifRS / Warnings / Warning	1..99	Used in conjunction with the Success element to define a business error.
@Type	1	Refers to OpenTravel Alliance EWT list (error warning type).
@Code	0..1	Refers to OpenTravel Alliance list ERR. Should be used wherever possible.
@RecordID	0..1	If the receiving system is able to identify within a batch of availability status messages which specific message failed, the UniqueID of the message should be reported here.
OTA_HotelAvailNotifRS / Errors	0..1	Indicates an error occurred during the processing of an OpenTravel message. If the message successfully processes, but there are business errors, those errors should be passed in the warning element.
OTA_HotelAvailNotifRS / Errors / Error	1..99	An error that occurred during the processing of a message.
@Type	1	Mandatory in OTA. Refers to OpenTravel Alliance EWT list (error warning type).
@Code	0..1	Optional. Refers to OpenTravel Alliance list ERR. Should be used wherever possible.
@RecordID	0..1	If the receiving system is able to identify within a batch of availability status messages which specific message failed, the UniqueID of the message should be reported here.

## 4.2 Get Availability

### 4.2.1 Data Element Table – Get Availability Request

Element  @Attribute	Num	Description/Contents
OTA_HotelAvailGetRQ	1	Root element of the message.
@EchoToken	0..1	A reference for additional message identification, assigned by the requesting host system. When a request message includes an echo token the corresponding response message MUST include an echo token with an identical value.
@TimeStamp	1	Timestamp of the transaction.
@Version	1	Version is a mandatory attribute in OTA. Therefore it must remain in HTNG in order to be able to use the same message.
OTA_HotelAvailGetRQ / POS / Source	0..1	Optional. This holds details regarding the requestor. It may be repeated to also accommodate the delivery systems.
OTA_HotelAvailGetRQ / POS / Source / RequestorID	1	The system sending the request message. An identifier of the entity making the request (e.g.; ATA/IATA/ID number, Electronic Reservation Service Provider (ERSP), Association of British Travel Agents (ABTA)).
@Type	1	Type attribute uses the OpenTravel Alliance code list UIT. For example, type 22 – ERSP (Electronic reservation service provider).
@ID	1	ID is used for the name or ID of the requestor. For example, Passkey.
OTA_HotelAvailGetRQ / HotelAvailRequests	1	Container for the individual HotelAvailRequests(s).
OTA_HotelAvailGetRQ / HotelAvailRequests / HotelAvailRequest	1	Provides the criteria to specify availability details to return. The criteria can include date ranges, room and products, rate plans, hurdle rate, etc. Optional. Although at least one should be sent for the message to be meaningful. Criteria and upper limit to be defined by trading partners.
@SendBookingLimit	0..1	Sending system requesting the booking limit to be returned.
@BookingLimitMessageType	0..1	Sending system requesting the booking limit of a particular type to be returned.
OTA_HotelAvailGetRQ / HotelAvailRequests / HotelAvailRequest / DateRange	1	A date or range of dates for the request criteria.
@Start, @End, @Duration	1	The first and last dates for the availability request. Duration is optional.
OTA_HotelAvailGetRQ / HotelAvailRequests / HotelAvailRequest / RatePlanCandidates	0..n	Collection of requested rate plans.

Element  @Attribute	Num	Description/Contents
OTA_HotelAvailGetRQ / HotelAvailRequests / HotelAvailRequest / RatePlanCandidates / RatePlanCandidate	0..n	Element used to specify products/rates.
@RatePlanCode	0..1	Conditionally mandatory if element is present. The rate plan for the availability request.
@RatePlanCategory	0..1	The rate plan category (collection of Rate Plan codes) for the availability request (i.e.; Corporate, Discount, Group). If RatePlanCode exists RatePlanCategory is not necessary.
OTA_HotelAvailGetRQ / HotelAvailRequests / HotelAvailRequest / RatePlanCandidates / RatePlanCandidate / HotelRefs	1	Collection of hotel identifiers.
OTA_HotelAvailGetRQ / HotelAvailRequests / HotelAvailRequest / RatePlanCandidates / RatePlanCandidate / HotelRefs / HotelRef	1	Information to identify one or more hotels.
@ChainCode	0..1	Optional. The Chain code for the availability request.
@BrandCode	0..1	Optional. The Brand Code for the availability request.
@HotelCode	1	This is the code of the property whose availability is being requested.
OTA_HotelAvailGetRQ / HotelAvailRequests / HotelAvailRequest / RoomTypeCandidates	0..n	Collection of room stay candidates.
OTA_HotelAvailGetRQ / HotelAvailRequests / HotelAvailRequest / RoomTypeCandidates / RoomTypeCandidate	0..n	Element used to specify room products.
@RoomTypeCode	1..n	Conditionally mandatory if element is present. The room type code whose availability is being requested.
OTA_HotelAvailGetRQ / HotelAvailRequests / HotelAvailRequest / RestrictionStatusCandidates	0..n	A collection of Restriction Status values.

Element  @Attribute	Num	Description/Contents
OTA_HotelAvailGetRQ / HotelAvailRequests / HotelAvailRequest / RestrictionStatusCandida tes / RestrictionStatusCandida te	0..n	Availability request assigned to the criteria specified.
@Restriction	0..1	Conditionally mandatory if element is present. If no other restriction is sent (Status, min LOS), then the Restriction field must be sent. In the OpenTravel Alliance message the following enumerations are possible: Master, Arrival, Departure, NonGuarantee, TravelAgent. However the HTNG recommendation should be that only Departure and Arrival are used as they are the only ones that are meaningful. If one of the partners implementing the message does not support one of the two Recommended enumerations, the un-supported enumeration would be ignored by the receiving system. For instance if the transmitting system supports both arrival and departure restriction (for this type of message) but the recipient only supports departure restrictions, then the arrival restrictions would be ignored. Partners would need to agree on this during implementation.
@Status	0..1	Conditionally mandatory if the element is present. Enumeration; possible values = "Open," "Close," "ClosedOnArrival," "OnRequest." If one of the partners implementing the message does not support all the enumerations, the enumerations that are not supported should be mapped to supported enumerations by the receiving system. An unsupported enumeration would be ignored by the receiving system – therefore the type of status messages must be agreed upon implementation between the two partners.
OTA_HotelAvailGetRQ / HotelAvailRequests / HotelAvailRequest / LengthsOfStayCandidate s	0..n	Collection of Length of Stay elements. These LOS elements indicate what LOS restrictions are to be included in the response.
@FixedPatternLength	0..1	Optional. If both partners agree to send the Full Pattern LOS then this field is set to the length of the pattern string further inside the message. For example, this would be set to 7 when the string has seven characters, from LOS1 to LOS7+.

Element  @Attribute	Num	Description/Contents
OTA_HotelAvailGetRQ / HotelAvailRequests / HotelAvailRequest / LengthsOfStayCandidates / LengthOfStayCandidate	0..1	A collection of patterns defining allowable lengths of stay (LOS).
@MinMaxMessageType	0..1	MinMaxLengthOfStay set = SetMinLOS when sending MinLOS, the default setting. It can be set = FullPatternLOS if both partners support the Full Pattern Length of Stay as available in the OpenTravel Alliance message. Other values could be used based on partner agreement.
OTA_HotelAvailGetRQ / HotelAvailRequests / HotelAvailRequest / BestAvailableRateCandidate	0..1	Use to request Best Available Rate criteria.
@SendLengthOfStayTime	0..1	Request length of stay associated with Best Available Rate.
@SendRatePlanCode	0..1	Request Rate plans associated with Best Available Rate.
@SendAmount	0..1	Request rate value amount associated with Best Available Rate.
@CurrencyCode	0..1	Request currency code associated with Best Available Rate.
OTA_HotelAvailGetRQ / HotelAvailRequests / HotelAvailRequest / HurdleRateCandidate	0..1	Use to request Hurdle Rates.
@SendAmount	0..1	Request rate value amount associated with Hurdle Rate.
@CurrencyCode	0..1	Request currency code associated with Hurdle Rate.
OTA_HotelAvailGetRQ / HotelAvailRequests / HotelAvailRequest / DeltaCandidate		Use to request Delta.
@SendAmount	0..1	Request rate value amount associated with Delta Rate.
@CurrencyCode	0..1	Request currency code associated with Delta Rate.
OTA_HotelAvailGetRQ / HotelAvailRequests / HotelAvailRequest / HotelRef	1	Indicates the detail of hotel reference information.
@ChainCode	0..1	Optional. The Chain code for the availability request.
@BrandCode	0..1	Optional. The Brand Code for the availability request.
@HotelCode	1	This is the code of the property whose availability is being requested.

#### 4.2.2 Data Element Table – Get Availability Response

Element  @Attribute	Num	Description/Contents
OTA_HotelAvailGetRS	1	Root element of the message.

Element  @Attribute	Num	Description/Contents
@EchoToken	0..1	A reference for additional message identification, assigned by the requesting host system. When a request message includes an echo token the corresponding response message MUST include an echo token with an identical value.
@TimeStamp	1	Timestamp of the transaction.
@Version	1	Version is a mandatory attribute in OTA. Therefore it must remain in HTNG in order to be able to use the same message.
OTA_HotelAvailGetRS / Success	1	This is the annotation that the availability request was received successfully. It could be combined with warning messages if some of the messages in the batch had issues.
OTA_HotelAvailGetRS / Warnings	0..1	Used when a message has been successfully processed to report any warnings or business errors that occur.
OTA_HotelAvailGetRS / Warnings / Warning	1..n	Used in conjunction with the Success element to define a business error.
@Type	1	Refers to OpenTravel Alliance EWT list (error warning type).
@Code	0..1	Optional. Refers to OpenTravel Alliance list ERR. Should be used wherever possible.
@RecordID	0..1	If the receiving system is able to identify within a batch of availability status messages which specific message failed, the UniqueID of the message should be reported here.
OTA_HotelAvailGetRS / Errors	1	Errors are returned if the request was unable to be processed.
@Type	1	Mandatory in OTA. Refers to OpenTravel Alliance EWT list (error warning type).
@Code	0..1	Optional. Refers to OpenTravel Alliance list ERR. Should be used wherever possible.
@RecordID	0..1	If the receiving system is able to identify within a batch of availability status messages which specific message failed, the UniqueID of the message should be reported here.
OTA_HotelAvailGetRS / AvailStatusMessages		Container for the individual AvailStatusMessage(s). An OTA_HotelAvailGetRS contains the availability statuses for a single hotel. Hotel identification information are the attributes of this element.
@ChainCode	0..1	Optional.
@BrandCode	0..1	Optional.
@HotelCode	1	Hotel Code the request was for.
OTA_HotelAvailGetRS / AvailStatusMessages / AvailStatusMessage	1	The AvailStatusMessage. It is here that one indicates whether the inventory is opened, closed, closed on request, etc.
@BookingLimitMessageType	0..1	If requested mandatory. Enumerated values are used to indicate whether the booking limit sent in the transmission is used to set, adjust or delete the booking limit. (SetLimit, AdjustLimit, RemoveLimit).

Element  @Attribute	Num	Description/Contents
@BookingLimit	0..1	If requested mandatory. This is the maximum number of rooms for the InvTypeCode sent that can be booked. Although all restrictions are optional, at least one should be sent for the message to have any meaning.
OTA_HotelAvailGetRS / AvailStatusMessages / AvailStatusMessage / StatusApplicationControl	1	Information in what the AvailStatus Message applies to (i.e, the combination of inventory and rates codes) and the period of application.
@Start, @End	1	The first and last dates for which the availability update is being sent.
@RatePlanCode	1	This is the rate plan whose availability is being updated.
@RatePlanCategory	0..1	This is the Category or segment to which the rate whose availability is being updated belongs.
@Mon, @Tue, @Weds, @Thur, @Fri, @Sat, @Sun	0..1	The day of the week indicators are used to communicate to which days of the week the update pertains. If one is sent they must all be sent.
OTA_HotelAvailGetRS / AvailStatusMessages / AvailStatusMessage / LengthsOfStay	1	Collection of Length of Stay elements. These LOS elements indicate what LOS restrictions are to be added or removed. Some systems include this information directly with the Availability Status as opposed to the booking restriction.
@FixedPatternLength	0..1	If both partners agree to send the Full Pattern LOS then this field is set to the length of the pattern string further inside the message. For example, this would be set to 7 when the string has seven characters, from LOS1 to LOS7+.
OTA_HotelAvailGetRS / AvailStatusMessages / AvailStatusMessage / LengthsOfStay / LengthOfStay	0..n	Conditionally mandatory. If LengthsOfStay is sent then there must be at least one LengthOfStay element.
@Time, @TimeUnit	0..1	Optional. If sending MinLOS then TimeUnit set = Day and Time set to the MinLOS value.
@MinMaxMessageType	0..1	MinMaxLengthOfStay set = SetMinLOS when sending MinLOS, the default setting. It can be set = FullPatternLOS if both partners support the Full Pattern Length of Stay as available in the OpenTravel Alliance message. Other values could be used based on partner agreement.
OTA_HotelAvailGetRS / AvailStatusMessages / AvailStatusMessage / LengthsOfStay / LengthOfStay / LOS_Pattern	0..1	Used to define the LOS Pattern.

Element  @Attribute	Num	Description/Contents
@FullPatternLOS	0..1	If sending Full Pattern LOS then this is set to the string with 'Y' for the LOS that are open and 'N' for the LOS that are closed. The length of the string is indicated in @FixedPatternLength described above. For example, to open the odd LOS (LOS1, LOS3, LOS5 and LOS7) and close the even LOS (LOS2, LOS4 and LOS6) then the string would be "YNYNYNY."
OTA_HotelAvailGetRS / AvailStatusMessages / AvailStatusMessage / BestAvailableRates	1	A collection of the best rates available.
OTA_HotelAvailGetRS / AvailStatusMessages / AvailStatusMessage / BestAvailableRates / BestAvailableRate	0..n	This element is normally repeated for each length of stay. The maximum length of stay "n" is to be agreed by trading partners, though it normally it is 8. In the case where a single BAR value is applicable for all length of stays, the element is sent once.
@LengthOfStayTime	1	Length of stay. This is normally a number from 1 to "n" (defined above). In the case where a single BAR value is applicable for all length of stays, the value is zero.
@RatePlanCode	1	This is the rate code identifying the best available rate.
@Amount	0..1	Value of the best available rate. Sent to validate the RatePlanCode. The BAR amount to be used by the Reservation system for any LOS that is greater than the agreed maximum, will be the same value given for the maximum length of stay "n." Amount is a basic value and refers to the general charge of the Rate plan. It is not expected to convey differing values based on the number of Adults and/or Children occupying a room. If required, the trading partners may agree the implementation of a more granular level of rate value using the OTA_HotelRateAmountNotifRQ message.
@CurrencyCode	0..1	Currency of delta adjustment using ISO4217 codes. If the partner receiving the data does not support currency code for Hurdle rate, the currency code would be ignored.
@TaxInclusive	0..1	Boolean Flag indicating whether the amount sent as Best Available Rate includes Tax or not. In the event of there needing to be more detailed information, it will be sent using the OpenTravel Alliance Rate Plan messages previously adopted.
OTA_HotelAvailGetRS / AvailStatusMessages / AvailStatusMessage / HurdleRate	0..1	Restriction based on the minimum rate to be considered for availability, i.e., can sell weekend rate only if charging the hurdle rate or more.
@Amount	1	This is the Hurdle rate the user wishes to set.
@CurrencyCode	0..1	Currency of delta adjustment using ISO4217 codes. If the partner receiving the data does not support currency code for Hurdle rate, the currency code would be ignored.



Element  @Attribute	Num	Description/Contents
OTA_HotelAvailGetRS / AvailStatusMessages / AvailStatusMessage / Delta	0..1	Incremental amount added to the hurdle rate (e.g., Amount = 20 USD and ceiling = 5, the maximum Delta would be 100 USD).
@Amount	0..1	Amount of delta adjustment.
@CurrencyCode	0..1	Currency of delta adjustment using ISO4217 codes. If the partner receiving the data does not support currency code for Hurdle rate, the currency code would be ignored.
@Ceiling, @MaxSold	0..1	Limits on the application of the Delta.
OTA_HotelAvailGetRS / AvailStatusMessages / AvailStatusMessage / UniqueID	1	The unique identifier element allows the trading partners to uniquely identify each AvailStatusMessage for tracing of transactions.
@Type	1	Type refers to OpenTravel Alliance code list UIT – nr 16 = Reference. This is used to identify each single availability status message for error reporting purposes.
@ID	1	ID is a unique incremental number for each availability message that identifies that specific message.
OTA_HotelAvailGetRS / AvailStatusMessages / AvailStatusMessage / RestrictionStatus	0..1	Availability status assigned to the room rate combination.
@Restriction	0..1	Optional. If no other restriction is sent (Status, min LOS), then the Restriction field must be sent. In the OpenTravel Alliance message the following enumerations are possible: Master, Arrival, Departure, NonGuarantee, TravelAgent. However, the HTNG recommendation should be that only Departure and Arrival are used as they are the only ones that are meaningful. If one of the partners implementing the message does not support one of the two Recommended enumerations, the unsupported enumeration would be ignored by the receiving system. For instance if the transmitting system supports both arrival and departure restriction (for this type of message) but the recipient only supports departure restrictions, then the arrival restrictions would be ignored. Partners would need to agree on this during implementation.
@Status	1	Enumeration; possible values = "Open," "Close," "ClosedOnArrival," "OnRequest." If one of the partners implementing the message does not support all the enumerations, the enumerations that are not supported should be mapped to supported enumerations by the receiving system. An unsupported enumeration would be ignored by the receiving system – therefore the type of status messages must be agreed upon implementation between the two partners.

## 4.3 Update Inventory

### 4.3.1 Data Element Table – Update Inventory Request

Element  @Attribute	Num	Description/Contents
OTA_HotelInvCountNotifRQ	1	Root element of the message.
@EchoToken	0..1	A reference for additional message identification, assigned by the requesting host system. When a request message includes an echo token the corresponding response message MUST include an echo token with an identical value.
@TimeStamp	1	Timestamp of the transaction.
@Version	1	Version is a mandatory attribute in OTA. Therefore it must remain in HTNG in order to be able to use the same message.
OTA_HotelInvCountNotifRQ / POS / Source	1	Optional. This holds details regarding the requestor. It may be repeated to also accommodate the delivery systems.
OTA_HotelInvCountNotifRQ / POS / Source / RequestorID	1	The system sending the request message. An identifier of the entity making the request (e.g. ATA/IATA/ID number, Electronic Reservation Service Provider (ERSP), Association of British Travel Agents (ABTA)).
@Type	1	Type attribute uses the OpenTravel Alliance code list UIT. For example, type 22 – ERSP (Electronic reservation service provider).
@ID	1	ID is used for the name or ID of the requestor. For example, SynXis.
OTA_HotelInvCountNotifRQ / Inventories	1	A collection of Inventory. Hotel identification information are the attributes of this element.
@ChainCode	0..1	Optional. The Chain code for which the availability is requested.
@BrandCode	0..1	Optional. The Brand Code for which the availability is requested.
@HotelCode	1	This is the code of the property whose availability is being requested.
OTA_HotelInvCountNotifRQ / Inventories / Inventory	1..n	This is the inventory information for a given rate plan, room type, date, etc.
OTA_HotelInvCountNotifRQ / Inventories / Inventory / StatusApplicationControl	0..n	Information on what the InvCountNotif Message applies to (i.e. the combination of inventory and/or rate codes) and the period of application.
@Start, @End	1	The first and last dates for which the availability update is being sent.
@Mon,@Tue,@Weds,@Thur, @Fri,@Sat,@Sun	0..1	The day of the week indicators are used to communicate which days of the week the update pertains to. If one is sent they must all be sent.
@AllInvCode	0..1 *	When true indicates that the data transmitted is for total inventory in the hotel. This field is required if @InvTypeCode is not sent.
@InvTypeCode	0..1 *	This is the room type code for which the update is being sent. This field is required if @AllInvCode="false" or missing. This field should not be sent if @AllInvCode="true."
@InvBlockCode	0..1	Code that identifies an inventory block.
@RatePlanCode	0..1	Code that identifies the rate plan associated with an inventory block.

Element  @Attribute	Num	Description/Contents
OTA_HotelInvCountNotifRQ / Inventories / Inventory / InvCounts	0..1	A collection of inventory counts.
OTA_HotelInvCountNotifRQ / Inventories / Inventory / InvCounts / InvCount	1..n	Individual inventory count (e.g.; Physical, Available, Sold, OOO, NAFS).
@CountType	1	This identifies the type of inventory count being reported. Refer to OpenTravel Code List Inventory Count Type (INV).  1           Physical 2           Definitive availability 3           Tentative availability 4           Definite sold 6           Out of order 8           Out of inventory 16          Group authorized 17          Group sold 18          Group available
@Count	1	Number of rooms of this type available for sale.
OTA_HotelInvCountNotifRQ / Inventories / Inventory / UniqueID	0	Use in response message in the case of an error to indicate which node failed.
@Type	0	Type refers to OpenTravel Alliance code list UIT – nr 16 = Reference. This is used to identify each single availability status message for error reporting purposes.
@ID	0	ID is a unique incremental number for each availability message that identifies that specific message.

#### 4.3.2 Data Element Table – Update Inventory Response

Element  @Attribute	Num	Description/Contents
OTA_HotelInvCountNotifRS	1	Root element of the message.
@EchoToken	0..1	A reference for additional message identification, assigned by the requesting host system. When a request message includes an echo token the corresponding response message MUST include an echo token with an identical value.
@TimeStamp	1	Timestamp of the transaction.
@Version	1	Version is a mandatory attribute in OTA. Therefore it must remain in HTNG in order to be able to use the same message.
OTA_HotelInvCountNotifRS / Success	1	This is the annotation that the availability request was received successfully. It could be combined with warning messages if some of the messages in the batch had issues.
OTA_HotelInvCountNotifRS / Warnings	1..n	Used when a message has been successfully processed to report any warnings or business errors that occurred.
@Type	1	Refers to OpenTravel Alliance EWT list (error warning type).

Element  @Attribute	Num	Description/Contents
@Code	0..1	Optional. Refers to OpenTravel Alliance list ERR. Should be used wherever possible.
@RecordID	0..1	If the receiving system is able to identify within a batch of availability status messages which specific message failed, the UniqueID of the message should be reported here.
OTA_HotelInvCountNotifRS / Errors	1	Errors are returned if the request was unable to be processed.
@Type	1	Mandatory in OTA. Refers to OpenTravel Alliance EWT list (error warning type).
@Code	0..1	Optional. Refers to OpenTravel Alliance list ERR. Should be used wherever possible.
@RecordID	0..1	If the receiving system is able to identify within a batch of availability status messages which specific message failed, the UniqueID of the message should be reported here.

## 5 Appendices

### 5.1 Glossary of Terms

For the purpose of this document the following terms have been defined as follows:

Term	Definition
Availability Publisher / Availability Control System	A system that maintains availability and notifies Availability Subscriber(s) of changes.
Availability Requester	A system that has the need to obtain availability information.
Availability Responder	A system that maintains availability information.
Availability Subscriber	A system that requires notification of availability changes as they occur in Availability Publisher system.
Inventory Publisher	A system that maintains inventory and notifies Inventory Subscriber(s) of changes.
Inventory Subscriber	A system that requires notification of inventory changes as they occur in Inventory Publisher system.
Extensible Markup Language (XML)	A general-purpose markup language for creating special-purpose markup languages, capable of describing different kinds of data.
Simple Object Access Protocol (SOAP)	A protocol specification for exchanging structured information in the implementation of Web Services in computer networks.

### 5.2 Implementation Notes

#### 5.2.1 *Errors and Warnings Usage*

The response messages have an option between sending a Success element with an Optional collection of Warning elements or a collection of Error elements. It has been agreed that when the request message is not processed the response will only have error elements. When the message is processed then the Success element will be sent, along with any warnings indicating issues that did not prevent the processing but should result in some future correction by the implementers of the message. Every AvailStatusMessage element in the request will be assigned a unique identifier. In the cases where an AvailStatusMessage element caused the error or the warning, the value of attribute RecordId will be set to match the unique identifier to connect the two. The intent is that technical support from either party can use the additional information to identify the cause of the problem. It should be noted that there may be cases where the error or warning may be caused at a higher level than the AvailStatusMessage and the value of RecordId cannot be set.

#### 5.2.2 *Min/Max LOS*

The method listed in the recommendations is the most basic method of passing Min/Max length of stay and therefore it should be the recommended way. Full LOS patterns can be sent

via the OpenTravel Alliance message and upon agreement between the implementing partners this method could be the one used instead of the recommended one.

### 5.3 Referenced Documents

The following table shows the documents upon which this document depends:

Document Title	Location/URL
Other HTNG Product Distribution specifications	Most recent versions outlined on <a href="#">workgroup's wiki page</a>
OpenTravel Alliance Specifications	<a href="http://www.opentravel.org/specifications">http://www.opentravel.org/specifications</a>