Protocol & Message Transport Specifications
Release 2011A

Event Notification Specification Version 1.0
22 April 2011

Protocol & Message Transport Workgroup

About HTNG

Hotel Technology Next Generation ("HTNG") is a nonprofit organization with global scope, formed in 2002 to facilitate the development of next-generation, customer-centric technologies to better meet the needs of the global hotel community. HTNG's mission is to provide leadership that will facilitate the creation of one (or more) industry solution set(s) for the lodging industry that:

- Are modeled around the customer and allow for a rich definition and distribution of hotel products, beyond simply sleeping rooms;
- Comprise best-of-breed software components from existing vendors, and enable vendors to collaboratively produce world-class software products encompassing all major areas of technology spending: hotel operations, telecommunications, in-room entertainment, customer information systems, and electronic distribution;
- Properly exploit and leverage a base system architecture that provides integration and interoperability through messaging; and that provides security, redundancy, and high availability;
- Target the needs of hotel companies up to several hundred properties, that are too small to solve the issues themselves;
- Will reduce technology management cost and complexity while improving reliability and scalability; and
- Can be deployed globally, managed remotely, and outsourced to service providers where needed.

In June 2005, HTNG announced the first-ever "Branding and Certification Program" for hotel technology. This program will enable vendors to certify their products against open HTNG specifications, and to use the "HTNG Certified" logo in their advertising and collateral materials.

It will enable hotels to determine which vendors have completed certification of their products against which specific capabilities, and the environments in which performance is certified. HTNG's vision is to achieve a flexible technical environment that will allow multiple vendors' systems to interoperate and that will facilitate vendor alliances and the consolidation of applications, in order to provide hotels with easily managed, continually evolving, cost-effective solutions to meet their complete technology needs on a global basis.

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.

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.

Table of Contents

1 Document History

1.1 Functional Change Log

Version	Date	Comments
1.0	22 April 2011	V1 functionality included: Basic Messages and First Draft of document

2 Acknowledgements

HTNG gratefully acknowledges the contributions of the following people in the development of this document:

Workgroup Member	Company
Chii Der Moo	FCS Computer Systems
Chris Laffoon	IBM Corporation
Jeremy Keller	MeetingMatrix International
Sophie Huang	Riverbed Solutions
Bill Rosenberg	Starwood Hotels & Resorts Worldwide

3 Document Information

3.1 Document Purpose

Systems in hospitality exchange data are at an ever increasing rate. Effective collaboration between systems means that more automated mechanisms need to be in place, thereby allowing data to be exchanged when key events occur (i.e. Guest Check-in, Guest Check-out, Room out of service).

This document provides details on a new series of messages that simplify the process of publicizing the messages that are available for subscription, along with an automated subscription and notification process.

3.2 Scope

This document provides the detail on a number of new messages along with an update to the HTNG Framework 2.1 header to provide a Publish and Subscribe methodology within workgroup messaging.

3.3 Audience

The intended audiences of this document are development teams and system designers seeking to implement these standardized interface specifications within their products. This document also provides business process flows that may used by hotel groups looking to standardize their interfaces within their hotel architectures.

3.4 Overview

The following new processes are supported with this set of messaging:

Activity	**Mechanism**
Show the event notification/subscriptions that are available	New messages – HTNG_SubscriptionsAvailableRQ/RS
Subscribe to a particular event notification	New messages – HTNG_SubscribeRQ/RS
Show the event notifications that my system is subscribed to	New messages – HTNG_SubscriptionStatusRQ/RS
Receive notifications on specific events	Notification section in HTNG Header along with transmission of Event message

3.5 Document Terms

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

Term	**Definition**
PMS	Property Management System
OpenTravel	OpenTravel Alliance (OTA) (http://www.opentravel.org)

3.6 Referenced Documents

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

Name	**Location**
OpenTravel Alliance Specifications	http://www.opentravel.org

4 Business Process

4.1 Overview

Many systems need to communicate key data to one another in order to support the business of hospitality. This technical specification provides a mechanism that allows systems to communicate in a more automated fashion.

An interested event consuming system can use the messages in this specification to:

- Determine which events have notification messages
- Subscribe to those notification messages
- Receive notification messages
- Determine which subscriptions are currently in place and modify the subscriptions as required

An interested event providing system can use the messages in this specification to:

- Publish the events that are available for subscription
- Receive requests for subscription
- Notify the appropriate systems when events take place
- Provide details on the events that are currently subscribed to

These messages allow the setup of the systems operating in the hospitality space to be far more collaborative. It also allows those designing software to move to decision for system interplay into the users control instead of the installer/administrator.

4.2 Roles

This specification defines the following roles:

4.2.1 Event Producer

A producer of event information that wishes to keep all interested systems updated on subscribed events.

4.2.2 Event Consumer

A system that wishes to consume information relating to events on a regular basis.

4.3 Message Flows

For clarity, the use cases described hereafter are listed in chronological order.

4.3.1 SubscriptionsAvailable

Provides a mechanism for a subscribing system to request a list of the available notification events that can it may subscribe to. This list includes the concept of versioned messages. This takes into account that various interested groups/vendors may need specific modifications to particular messages to support unique business functions.

4.3.2 Subscription

Provides a mechanism for a system to subscribe to specific events. The subscription process includes a unique ID from the subscribing system called the ConsumerSubscriptionID. This provides a mechanism for the subscribed system to update its subscription request using an overlay methodology.

4.3.3 SubscriptionStatus

Provides a mechanism to obtain a list of the existing subscriptions that a consuming system may have in place.

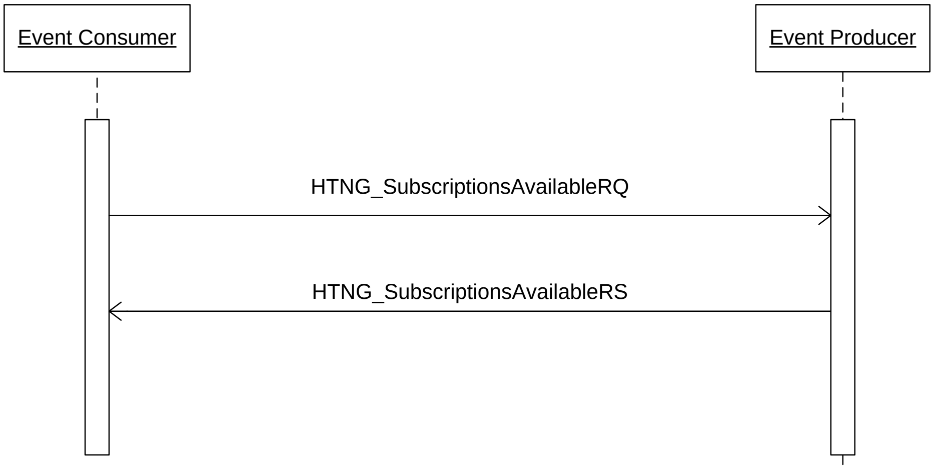
4.3.4 Notification

The process that is used to notify Event Consumers (subscribed systems) when an event notification takes place.

5 Use Cases

5.1 SubscriptionsAvailable

5.1.1 Messaging Use Case



Use Case Name:	**Subscriptions Available**
Summary:	A potential subscriber sends a subscriptions available request to a notification producer. The notification producer sends the set of events available for subscription.
Basic Course of Events:	The use case begins when the interested system (Event Consumer) requests a list of all of the events that are able to be subscribed to that are available from a producer system. 1. The EventConsumer issues an HTNG_SubscriptionsAvailableRQ 2. The Event Producer returns a payload that includes all of the messages that may be subscribed to along with the Producer Endpoints where the subscription must be submitted. 3. The Event Consumer is assumed to have a working knowledge of the messages that may be available for subscription.
Exception Path:	If no subscriptions are available, then an empty set of data is returned.
Alternative Paths:	None.
Trigger:	The Event Consumer has some interest in the available

	messages from a producer system.
Assumptions:	1. Subscription capability exists on the producer system. 2. Messages exist that may be subscribed to on a producer system.
Preconditions:	The two systems that know about each other have the appropriate authentications in place and are able to communicate.
Postconditions:	

5.1.2 Data Element Table – Request

Element | @Attribute	**Num**	**Description/Contents**
HTNG_SubscriptionsAvailableRQ	1	Root element of the message. This message initiates the request for the subscriptions available from the producer system.
@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	0..1	Indicates the creation date and time of the message in UTC using the following format specified by ISO 8601; YYYY-MM-DDThh:mm:ssZ with time values using the 24-hour clock (e.g., 20 November 2003, 1:59:38 pm UTC becomes 2003-11-20T13:59:38Z).
@Version	1	For all OpenTravel versioned messages, the version of the message is indicated by a decimal value.
@Target	0..1	Used to indicate whether the request is for the Test or Production system. Value can be either 'Test' or 'Production'.
HTNG_SubscriptionsAvailableRQ / TPA_Extensions	0..1	Used for custom exchange of data agreed to by trading partners.

5.1.3 Sample Message – Request

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_SubscriptionsAvailableRQ EchoToken="0d25a8b6-bbfc-48cb-add1-e87672d3355e"
TimeStamp="2011-03-25T09:30:47Z" Version="1.0" Target="Production" xmlns="http://htng.org/2010B"
xmlns:ota="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
</HTNG_SubscriptionsAvailableRQ>
```

5.1.4 Data Element Table – Response

Element | @Attribute	Num	Description/Contents
HTNG_SubscriptionsAvailableRS	1	Root element of the message. This message provides all of the available subscriptions from the producer system.
@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	Indicates the creation date and time of the message in UTC using the following format specified by ISO 8601; YYYY-MM-DDThh:mm:ssZ with time values using the 24-hour clock (e.g., 20 November 2003, 1:59:38 pm UTC becomes 2003-11-20T13:59:38Z).
@Version	1	For all OpenTravel versioned messages, the version of the message is indicated by a decimal value.
@Target	0..1	Used to indicate whether the request is for the Test or Production system.
HTNG_SubscriptionsAvailableRS / Success	0..1	The presence of the empty Success element explicitly indicates that the OpenTravel versioned message succeeded.
HTNG_SubscriptionsAvailableRS / Warnings	0..1	Used in conjunction with the Success element to define one or more business errors.
HTNG_SubscriptionsAvailableRS / Warnings / Warning	1..n	Used when a message has been successfully processed to report any warnings or business errors that occurred.
@Type	1	The Warning element MUST contain the Type attribute that uses a recommended set of values to indicate the warning type. The validating XSD can expect to accept values that it has NOT been explicitly coded for and process them by using Type ="Unknown". Refer to OpenTravel Code List Error Warning Type (EWT).
@Status	0..1	If present, recommended values are those enumerated in the OTA_ErrorRS, (NotProcessed | Incomplete | Complete | Unknown) however, the data type is designated as string data, recognizing that trading partners may identify additional status conditions not included in the enumeration.
@ShortText	1	An abbreviated version of the error in textual format.
@Code	0..1	If present, this refers to a table of coded values exchanged between applications to identify errors or warnings. Refer to

Element | @Attribute	Num	Description/Contents
		OpenTravel Code List Error Codes (ERR).
HTNG_SubscriptionsAvailableRS / AvailableEventTypes	0..1	Provides the Event Types that a system offers as possible subscriptions
HTNG_SubscriptionsAvailableRS / AvailableEventTypes / AvailableEventType	1..n	Each individual Event Type the publishing system is making available for subscription.
@ProducerEventID		A unique Identifier provided by the Producer to define a specific message that can be subscribed to. For the same MessageEventType, there may be many ProducerEventIDs due to generational and specific changes made for certain interested groups/vendors.
@AvailableSince	0..1	Defines when the subscription became available.
@VendorVersionNumber	0..1	Provides a sequential number of the revision to this particular message when an event is updated along with a VendorProfileID. This allows for different subscriptions/content for different vendors. Vendors are strongly encouraged to formalize this subscription scenario in future HTNG schema releases.
@VendorProfileID	0..1	Along with the VendorVersionID, this uniquely defines a particular genus of messages per interested group/vendor.
@MessageEventType	1	Defines the Event Type, typically the message name (i.e. OTA_HotelResNotifRQ). This should translate directly to the messages defined by workgroups.
@ProducerReason	1	The source/reason/trigger that caused the event to be published (for instance, check-in, check-out, room move, etc.). The values represented should be mutually agreed upon between vendors.
HTNG_SubscriptionsAvailableRS / AvailableEventTypes / AvailableEventType / ProducerEndpoint	0..1	Provides the producer endpoint that should be used when subscribing to this message. A producer may have multiple endpoints to deal with multiple group/vendor specific modifications in place (see also VendorProfileID and VendorVersionID).
HTNG_SubscriptionsAvailableRS / AvailableEventTypes / AvailableEventType / TargetNamespace	0..1	Describes the namespace of the message that the consumer will be receiving. For instance, messages defined in the HTNG 2011A namespace will have a value of http://htng.org/2011A and OpenTravel messages will have a value of http://www.opentravel.org/OTA/2003/05.
HTNG_SubscriptionsAvailableRS / AvailableEventTypes /	0..1	Brief description of the Event Type. More detail should be available from the various message specifications within

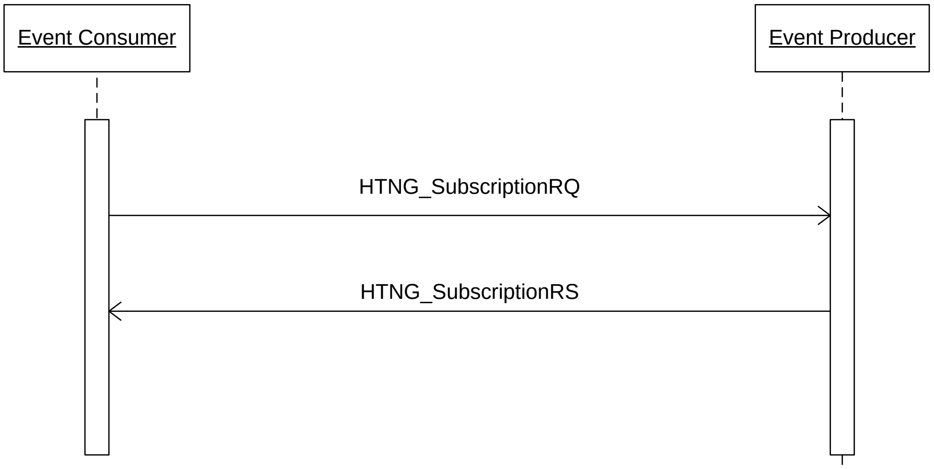
Element | @Attribute	**Num**	**Description/Contents**
AvailableEventType / Description		other workgroups.
HTNG_SubscriptionsAvailableRS / AvailableEventTypes / AvailableEventType / AvailableFilters	0..1	Describes the filters that may be available for a specific Event Type.
HTNG_SubscriptionsAvailableRS / AvailableEventTypes / AvailableEventType / AvailableFilters / Filter	1..n	A filter expression negotiated between trading partners. Future versions may dictate a specific structure.

5.1.5 Sample Message – Response

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_SubscriptionsAvailableRS EchoToken="0d25a8b6-bbfc-48cb-add1-e87672d3355e"
TimeStamp="2011-03-25T09:30:49Z" Version="1.0" Target="Production"
xmlns="http://htng.org/2011A" xmlns:ota="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <Success/>
      <AvailableEventTypes>
            <AvailableEventType ProducerEventID="06997b02-4f43-46cf-978a-c7dee7fd2c3b"
AvailableSince="2010-12-117T14:23:58Z" VendorVersionID="1.2" VendorProfileID=""
MessageEventType="OTA_HotelResNotifRQ" ProducerReason="NewRes">

      <ProducerEndpoint>http://gateway.eventproducersystem.com/HTNG_PublishAndSubscribeS
ervice/HTNG_PublishAndSubscribeService.wsdl</ProducerEndpoint>

      <TargetNamespace>http://www.opentravel.org/OTA/2003/05</TargetNamespace>
                  <Description>Publishes and event whenever a new reservation is
created</Description>
            </AvailableEventType>
            <AvailableEventType ProducerEventID="bae3f855-36a6-4dad-a433-9b8c58e514e7"
AvailableSince="2010-12-117T14:25:12Z" VendorVersionID="1.7" VendorProfileID=""
MessageEventType="OTA_HotelResNotifRQ" ProducerReason="ResMod">

      <ProducerEndpoint>http://gateway.eventproducersystem.com/HTNG_PublishAndSubscribeS
ervice/HTNG_PublishAndSubscribeService.wsdl</ProducerEndpoint>

      <TargetNamespace>http://www.opentravel.org/OTA/2003/05</TargetNamespace>
                  <Description>Publishes and event whenever a reservation is
modified</Description>
            </AvailableEventType>
            <AvailableEventType ProducerEventIDe61b7790-5790-4150-97cb-5a2e87c88e2c"
AvailableSince="2010-12-117T14:20:34Z" VendorVersionID="1.3" VendorProfileID=""
MessageEventType="OTA_CancelRQ" ProducerReason="ResCxl">

      <ProducerEndpoint>http://gateway.eventproducersystem.com/HTNG_PublishAndSubscribeS
ervice/HTNG_PublishAndSubscribeService.wsdl</ProducerEndpoint>

      <TargetNamespace>http://www.opentravel.org/OTA/2003/05</TargetNamespace>
                  <Description>Publishes and event whenever a reservation is
canceled</Description>
            </AvailableEventType>
      </AvailableEventTypes>
</HTNG_SubscriptionsAvailableRS>
```

5.2 Subscription

5.2.1 Messaging Use Case



Use Case Name:	**Subscription**
Summary:	A vendor system sends a subscription request to the notification producer to register interest for an event type. The notification system confirms that the subscription has been received and created within its notification register.
Basic Course of Events:	The use case begins when the interested system (Event Consumer) issues a request to subscribe to a particular event notification that was detailed in the HTNG_SubscriptionsAvailableRS message.
Exception Path:	If the subscription request is not valid, then a fault will be returned (i.e. the subscription request was for a ProducerEventID that does not exist).
Alternative Paths:	None.
Trigger:	The Event Consumer wishes to subscribe to an event.
Assumptions:	Subscription capability exists on the producer system. An HTNG_SubscriptionsAvailableRS message has provided some ProducerEventIDs that are available.
Preconditions:	The two systems that know about each other have the appropriate authentications in place and are able to communicate.

Postconditions:

5.2.2 Data Element Table – Request

Element | @Attribute	Num	Description/Contents
HTNG_SubscriptionRQ	1	Root element of the message. The main message used by the subscribing system to subscribe to a particular Event Type. The primary element (ProducerEventID) ensures uniqueness of the request. Note: To delete an existing subscription, @TerminationDateTime should be set to a date/time in the past, effectively expiring the subscription immediately.
@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	0..1	Indicates the creation date and time of the message in UTC using the following format specified by ISO 8601; YYYY-MM-DDThh:mm:ssZ with time values using the 24-hour clock (e.g., 20 November 2003, 1:59:38 pm UTC becomes 2003-11-20T13:59:38Z).
@Version	1	For all OpenTravel versioned messages, the version of the message is indicated by a decimal value.
@Target	0..1	Used to indicate whether the request is for the Test or Production system. Value can be either 'Test' or 'Production'.
HTNG_SubscriptionRQ / EventType	1	Defines the particular class of Event that is being subscribed to.
@ConsumerSubscriptionID	0..1	The Subscriber's unique reference ID for this subscription. If this is sent in as a duplicate, then the producing system should update/overlay all parameters/elements selected (i.e. ConsumerEndpoint).
@ProducerEventID	1	A unique Identifier provided by the Producer to define a specific message that can be subscribed to. For the same MessageEventType there may be many ProducerEventIDs due to generational and specific changes made for certain interested groups/vendors.
@TerminationDateTime	0..1	A date/time that details when a subscription should end.
HTNG_SubscriptionRQ / EventType / ConsumerEndpoint	1	The URL of the Notification Consumers listener where Events should be sent.
HTNG_SubscriptionRQ /	0..1	Describes the filters being requested for a specific Event Type.

Element | @Attribute	Num	Description/Contents
EventType / RequestedFilters		
HTNG_SubscriptionRQ / EventType / RequestedFilters / Filter	1..n	A filter expression negotiated between trading partners. Future versions may dictate a specific structure.
HTNG_SubscriptionRQ / EventType / TPA_Extensions	0..1	Used for custom exchange of data agreed to by trading partners.

5.2.3 Sample Message – Request

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_SubscriptionRQ EchoToken="5df816e4-fe2d-4902-bebe-404468e8f3d2" TimeStamp="2011-
03-25T09:31:18Z" Version="1.0" Target="Production" xmlns="http://htng.org/2011A"
xmlns:ota="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <EventType ConsumerSubscriptionID="05be67ac-bb98-42c4-aa79-41d121526aa9"
ProducerEventID="06997b02-4f43-46cf-978a-c7dee7fd2c3b" TerminationDateTime="2012-03-
25T09:31:18Z">
      <ConsumerEndpoint>http://webservices.consumersystem.com/HTNG_ARIAndReservationPush
Service/HTNG_ARIAndReservationPushService.wsdl</ConsumerEndpoint>
      </EventType>
</HTNG_SubscriptionRQ>
```

5.2.4 Data Element Table – Response

Element | @Attribute	Num	Description/Contents
HTNG_SubscriptionRS	1	Root element of the message. The response confirming the subscription has been created, updated or expired.
@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	Indicates the creation date and time of the message in UTC using the following format specified by ISO 8601; YYYY-MM-DDThh:mm:ssZ with time values using the 24-hour clock (e.g., 20 November 2003, 1:59:38 pm UTC becomes 2003-11-20T13:59:38Z).
@Version	1	For all OpenTravel versioned messages, the version of the message is indicated by a decimal value.
@Target	0..1	Used to indicate whether the request is for the Test or Production system.
HTNG_SubscriptionRS / Success	0..1	The presence of the empty Success element explicitly indicates that the OpenTravel versioned message succeeded.
HTNG_SubscriptionRS /	0..1	Used in conjunction with the Success element to define one or

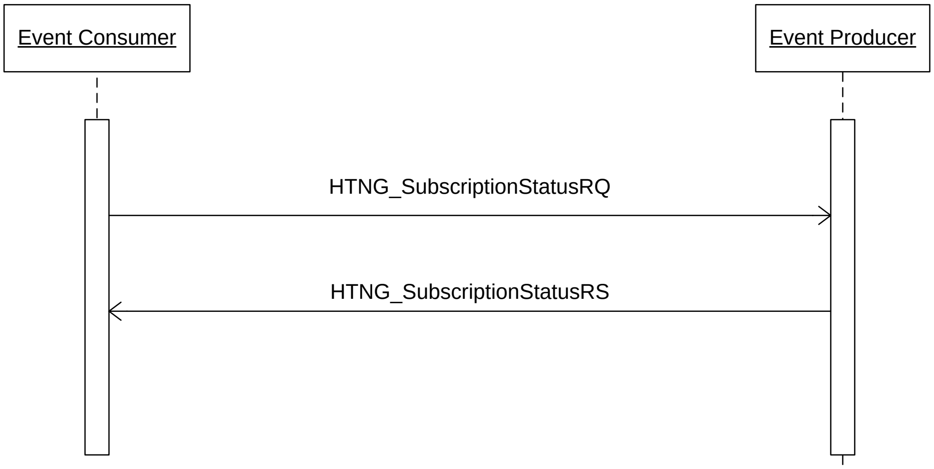
Element | @Attribute	Num	Description/Contents
Warnings		more business errors.
HTNG_SubscriptionRS / Warnings / Warning	1..n	Used when a message has been successfully processed to report any warnings or business errors that occurred.
@Type	1	The Warning element MUST contain the Type attribute that uses a recommended set of values to indicate the warning type. The validating XSD can expect to accept values that it has NOT been explicitly coded for and process them by using Type ="Unknown". Refer to OpenTravel Code List Error Warning Type (EWT).
@Status	0..1	If present, recommended values are those enumerated in the OTA_ErrorRS, (NotProcessed | Incomplete | Complete | Unknown) however, the data type is designated as string data, recognizing that trading partners may identify additional status conditions not included in the enumeration.
@ShortText	1	An abbreviated version of the error in textual format.
@Code	0..1	If present, this refers to a table of coded values exchanged between applications to identify errors or warnings. Refer to OpenTravel Code List Error Codes (ERR).

5.2.5 Sample Message – Response

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_SubscriptionRS EchoToken="5df816e4-fe2d-4902-bebe-404468e8f3d2" TimeStamp="2011-
03-25T09:31:20Z" Version="1.0" Target="Production" xmlns="http://htng.org/2011A"
xmlns:ota="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <Success/>
</HTNG_SubscriptionRS>
```

5.3 SubscriptionStatus

5.3.1 Messaging Use Case



Use Case Name:	**SubscriptionStatus**
Summary:	The subscribing system is interested in determining which types of events it is subscribed to. It sends a request for the list of its current subscriptions. The producer system replies with a list of subscribed event types.
Basic Course of Events:	The use case begins when the interested system (Event Consumer) issues a request to provide the details for all of the subscriptions currently in place.
Exception Path:	If no subscriptions currently exist, an empty set of data is returned.
Alternative Paths:	None.
Trigger:	The EventConsumer wishes to know which events it is subscribed to.
Assumptions:	Subscription capability exists on the producer system The HTNG_SubscriptionStatusRQ/RS message pair is available and supported.
Preconditions:	The two systems that know about each other have the appropriate authentications in place and are able to communicate.
Postconditions:	

5.3.2 Data Element Table – Request

Element | @Attribute	Num	Description/Contents
HTNG_SubscriptionStatusRQ	1	Root element of the message. Requests the existing subscriptions currently in place for a particular Event Consumer.
@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	0..1	Indicates the creation date and time of the message in UTC using the following format specified by ISO 8601; YYYY-MM-DDThh:mm:ssZ with time values using the 24-hour clock (e.g., 20 November 2003, 1:59:38 pm UTC becomes 2003-11-20T13:59:38Z).
@Version	1	For all OpenTravel versioned messages, the version of the message is indicated by a decimal value.
@Target	0..1	Used to indicate whether the request is for the Test or Production system. Value can be either 'Test' or 'Production'.
HTNG_SubscriptionStatusRQ / TPA_Extensions	0..1	Used for custom exchange of data agreed to by trading partners.

5.3.3 Sample Message – Request

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_SubscriptionStatusRQ EchoToken="96a7816e-acfd-4803-aae4-4d859220fc03"
TimeStamp="2011-03-25T09:31:21Z" Version="1.0" Target="Production"
xmlns="http://htng.org/2011A" xmlns:ota="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
</HTNG_SubscriptionStatusRQ>
```

5.3.4 Data Element Table – Response

Element | @Attribute	Num	Description/Contents
HTNG_SubscriptionStatusRS	1	Root element of the message. Describes the current subscriptions in place with the Event Producer.
@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	Indicates the creation date and time of the message in UTC using the following format specified by ISO 8601; YYYY-MM-DDThh:mm:ssZ with time values using the 24-hour clock (e.g., 20 November 2003, 1:59:38 pm UTC becomes 2003-11-20T13:59:38Z).
@Version	1	For all OpenTravel versioned messages, the version of the

Element | @Attribute	Num	Description/Contents
		message is indicated by a decimal value.
@Target	0..1	Used to indicate whether the request is for the Test or Production system.
HTNG_SubscriptionStatusRS / Success	0..1	The presence of the empty Success element explicitly indicates that the OpenTravel versioned message succeeded.
HTNG_SubscriptionStatusRS / Warnings	0..1	Used in conjunction with the Success element to define one or more business errors.
HTNG_SubscriptionStatusRS / Warnings / Warning	1..n	Used when a message has been successfully processed to report any warnings or business errors that occurred.
@Type	1	The Warning element MUST contain the Type attribute that uses a recommended set of values to indicate the warning type. The validating XSD can expect to accept values that it has NOT been explicitly coded for and process them by using Type ="Unknown". Refer to OpenTravel Code List Error Warning Type (EWT).
@Status	0..1	If present, recommended values are those enumerated in the OTA_ErrorRS, (NotProcessed | Incomplete | Complete | Unknown) however, the data type is designated as string data, recognizing that trading partners may identify additional status conditions not included in the enumeration.
@ShortText	1	An abbreviated version of the error in textual format.
@Code	0..1	If present, this refers to a table of coded values exchanged between applications to identify errors or warnings. Refer to OpenTravel Code List Error Codes (ERR).
HTNG_SubscriptionStatusRS / Subscriptions	0..1	A collection of the events the Event Consumer is currently subscribed to.
HTNG_SubscriptionStatusRS / Subscriptions / Subscription	1..n	Child element describing each subscription currently in place.
@ConsumerSubscriptionID	1	The Subscriber's unique reference ID for this subscription. If this is sent in as a duplicate then the producing system should update/overlay all parameters/elements selected (i.e. ConsumerEndpoint).
@ProducerEventID	1	A unique Identifier provided by the Producer to define a specific message that can be subscribed to. For the same MessageEventType, there may be many ProducerEventIDs due to generational and specific changes made for certain interested groups/vendors.
@TerminationDateTime	0..1	A date/time that details when a subscription should end.

Element | @Attribute	**Num**	**Description/Contents**
@LastUpdatedDateTime	0..1	The date/time the subscription was last modified.
@VendorVersionID	0..1	Provides a sequential number of the revision to this particular message when an event is updated along with a VendorProfileID. This allows for different subscriptions/content for different vendors. Vendors are strongly encouraged to formalize this subscription scenario in future HTNG schema releases.
@VendorProfileID	0..1	Along with the VendorVersionID, this uniquely defines a particular genus of messages per interested group/vendor.
@MessageEventType	1	Defines the Event Type, typically the message name (i.e. OTA_HotelResNotifRQ). This should translate directly to the messages defined by workgroups.
@ProducerReason	1	The source/reason/trigger that caused the event to be published (for instance, check-in, check-out, room move, etc.). The values represented should be mutually agreed upon between vendors.
HTNG_SubscriptionStatusRS / Subscriptions / Subscription / ProducerEndpoint	1	Provides the producer endpoint that should be used when subscribing to this message. A producer may have multiple endpoints to deal with multiple group/vendor specific modifications in place (see also VendorProfileID and VendorVersionID).
HTNG_SubscriptionStatusRS / Subscriptions / Subscription / ConsumerEndpoint	1	The URL of the Notification Consumers listener where Events should be sent.
HTNG_SubscriptionStatusRS / Subscriptions / Subscription / TargetNamespace	0..1	Describes the namespace of the message the consumer will be receiving. For instance, messages defined in the HTNG 2011A namespace will have a value of http://htng.org/2011A and OpenTravel messages will have a value of http://www.opentravel.org/OTA/2003/05.
HTNG_SubscriptionStatusRS / Subscriptions / Subscription / Description	0..1	Brief description of the Event Type. More detail should be available from the various message specifications within other workgroups.
HTNG_SubscriptionStatusRS / Subscriptions / Subscription / AppliedFilters	0..1	Describes the filters being applied for a specific Event Type. In version 1 of the specification, this is a place holder.
HTNG_SubscriptionStatusRS / Subscriptions /	1..n	A filter expression negotiated between trading partners. Future versions may dictate a specific structure.

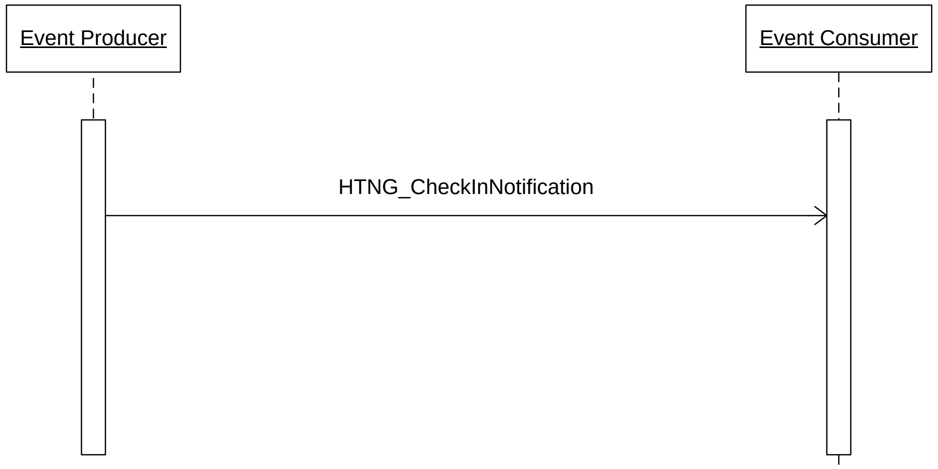
Element | @Attribute	Num	Description/Contents
Subscription / AppliedFilters / Filter		

5.3.5 Sample Message – Response

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_SubscriptionStatusRS EchoToken="96a7816e-acfd-4803-aae4-4d859220fc03"
TimeStamp="2011-03-25T09:31:22Z" Version="1.0" Target="Production"
xmlns="http://htng.org/2011A" xmlns:ota="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <Success/>
      <Subscriptions>
            <Subscription ConsumerSubscriptionID="05be67ac-bb98-42c4-aa79-41d121526aa9"
ProducerEventID="06997b02-4f43-46cf-978a-c7dee7fd2c3b" TerminationDateTime="2012-03-
25T09:31:18Z" LastUpdatedDateTime="2011-03-25T09:31:19Z" VendorVersionID="1.2"
VendorProfileID="" MessageEventType="OTA_HotelResNotifRQ" ProducerReason="NewRes">
                  <ProducerEndpoint>
http://gateway.eventproducersystem.com/HTNG_PublishAndSubscribeService/HTNG_PublishAndSu
bscribeService.wsdl</ProducerEndpoint>
                  <ConsumerEndpoint>
http://webservices.consumersystem.com/HTNG_ARIAndReservationPushService/
HTNG_ARIAndReservationPushService.wsdl</ConsumerEndpoint>

      <TargetNamespace>http://www.opentravel.org/OTA/2003/05</TargetNamespace>
                  <Description>Publishes and event whenever a new reservation is
created</Description>
            </Subscription>
      </Subscriptions>
</HTNG_SubscriptionStatusRS>
```

5.4 Notification

5.4.1 Messaging Use Case



Use Case Name:	**Notification**

Summary:	The producer system determines that an event notification is needed for particular subscribers. The appropriate events are then sent to the subscribers that are registered for that event.
Basic Course of Events:	The use case begins when the Event Producer determines that an event that has occurred is of particular interest to an Event Consumer.
Exception Path:	If the Event Consumer is no longer available at the nominated endpoint the Producer will store Event notification messages for a maximum of 5 days and then delete them.
Alternative Paths:	None.
Trigger:	The Event producer wishes to send out an Event Notification.
Assumptions:	Subscription capability exists on the Producer system.
Preconditions:	The two systems that know about each other have the appropriate authentications in place and are able to communicate.
Postconditions:	

5.4.2 Data Element Table – Request

Element | @Attribute	**Num**	**Description/Contents**
.. / SubscriptionReference	1	The root element in the SOAP Header that encapsulates the event information being broadcast.
.. / SubscriptionReference @SubscriptionEventType	1	Defines the Event Type, typically the message name (i.e. OTA_HotelResNotifRQ). This should translate directly to the messages defined by workgroups.
.. / SubscriptionReference @SubscriptionEventTypeID	1	A unique Identifier provided by the Producer to define a specific message that can be subscribed to. For the same MessageEventType, there may be many ProducerEventIDs due to generational and specific changes made for certain interested groups/vendors.
.. / SubscriptionReference @ProducerReason	1	The source/reason/trigger that caused the event to be published (for instance, check-in, check-out, room move, etc.). The values represented should be mutually agreed upon between vendors.
.. / SubscriptionReference @ConsumerSubscriptionID	1	The Subscriber's unique reference ID for this subscription. If this is sent in as a duplicate then the producing system should update/overlay all parameters/elements selected (i.e. ConsumerEndpoint).
.. / SubscriptionReference @TerminationDateTime	0..1	A date/time that details when a subscription should end.

Element | @Attribute	Num	Description/Contents
.. / SubscriptionReference / SubscriptionProducerMessages	0..1	A collection of messages the Event Producer would like the Event Consumer to be aware of.
.. / SubscriptionReference / SubscriptionProducerMessages / SubscriptionProducerMessage	1..n	Provides a mechanism to instruct the Consumer system of some impending matter (for instance, credentials that are due to expire).

5.4.3 Sample Message – Request

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ota="http://www.opentravel.org/OTA/2003/05" xmlns:htng="http://htng.org/2011A"
targetNamespace="http://htng.org/2011A">
      <soap:Header>
            <htng:SubscriptionReference SubscriptionEventType="OTA_HotelResNotifRQ"
SubscriptionEventTypeID="22e8a584-0d18-4228-b2a8-3716fa2097fa" ProducerReason"NewRes"
ConsumerSubscriptionID="05be67ac-bb98-42c4-aa79-41d121526aa9" TerminationDateTime="2012-
03-25T09:31:18Z">
                  <htng:SubscriptionProducerMessages>

      <htng:SubscriptionProducerMessage></htng:SubscriptionProducerMessage>
                  </htng:SubscriptionProducerMessages>
            </htng:SubscriptionReference>
      </soap:Header>
      <soap:Body>
            <ota:HTNG_HotelResNotifRQ>
                  …
            </ota:HTNG_HotelResNotifRQ>
      </soap:Body>
</soap:Envelope>
```

5.4.4 WSDL Modifications

Most of the WSDL modification necessary for consumers to receive event notifications have been included in the HTNG 2011A Artifacts distribution. However, the binding definition in the WSDL will need to be modified to allow the inclusion of the HTNG_EventNotificationHeader in the SOAP Header. The following line will need to be inserted as a child of the appropriate `definitions/binding/operation/input` element:

```
<soap:header message="tns:HTNG_EventNotificationHeader"
part="HTNG_EventNotificationHeader" use="literal"/>
```