BULK DATA API SPECIFICATION

Version 1.0

Publication Date: 14 November 2017

About HTNG

Hospitality 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 hospitality professionals 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 facilitate 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.

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.

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 Hospitality Technology Next Generation and HTNG, and logos depicting these names, are trademarks of Hospitality 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 Hospitality 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

1 This Specification at a Glance

Many systems need to distribute batches of key data from a central repository to other systems on a regular basis to support the business of hospitality. This technical specification provides a mechanism that allows systems to communicate in a more automated fashion.

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

- Determine which files are available for subscription
- Subscribe to receive those files on a regular basis
- Unsubscribe from receiving those files when no longer desired
- Determine which subscriptions are currently in place and modify the subscriptions as required
- Request status of existing subscriptions

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

- Return a list of files that are available for subscription
- Receive requests for subscription and unsubscribe to those files
- Distribute the files to subscribing systems on a regular basis
- Provide details on the existing file subscriptions
- Receive modifications to existing subscriptions
- End an existing subscription and notify the subscriber

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 the decisions for system interplay into the user's control instead of the installer/administrator.

2 Document Information

2.1 Document Purpose

Systems in hospitality exchange data at an ever-increasing rate. Effective collaboration between systems means more automated mechanisms need to be in place to allow batches of data to be exchanged at regular intervals, rather than when key events occur.

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

2.2 Scope

This document provides detail on a number of messages to provide a "Publish and Subscribe" methodology for transfer of bulk data via batches within workgroup messaging.

2.3 Relationship to Other Standards

This specification and its supporting schemas leverage the existing OpenTravel Alliance methodology for message construction and draw upon data definitions common to several HTNG specifications as of November 2017.

Related specifications:

- All HTNG specifications may be found here, including a quick start guide for developers: http://www.htng.org/page/SpecsbyProductType
- HTNG Event Notification Specification
- OpenTravel Alliance Specifications

2.4 Audience

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

2.5 General Description

The purpose of this specification is to provide a uniform, but flexible means to automatically exchange files of information between systems.

General applications that need to receive files will follow a process similar to the following:

1. Get a catalogue of files that are available for subscription by calling "getCatalogue."
2. Subscribe to the files that need to be sent on a regular basis.

3. The File Distribution System will send files based on the rules in the subscription and will also send a notification to allow the tracking of attempts to send a file.

When subscribing, several options need to be considered:

- Does the subscriber want the files automatically sent? If so, a URL to receive the files needs to be provided.
- Does the subscriber prefer to receive a notification that a file is ready and retrieve the file itself? Then, a notification endpoint URL needs to be provided instead of a URL to receive the files.

The status of a subscription may be checked by calling the “getSubscriptionStatus” operation and a subscription can be renewed by calling the “RenewSubscriptionRQ” message.

If the Subscription Manager needs to cancel a subscription it will send a “SubscriptionEnd” notification to the subscriber.

2.6 Roles

Role Name	**Definition**	**Example**
Subscriber	The system that subscribes to receive files	PMS systems needing to receive reservation modifications from the Central Reservation System
Subscription Manager	A web service that accepts requests to manage, gets the status of, renews and/or cancels subscriptions on behalf of a File Distribution System	The Integration Platform
File Distribution System (FDS)	The system responsible for distributing files	The Central Reservation System
File Receiver	The system that will receive the files	The PMS
File Retriever	A system that will retrieve one or more files based on a notification	Back Office Accounting systems

2.7 Known Limitations

There are no known limitations as of the writing of this document. The specification and messages assume the implementer will use well known means of file distribution, such as secure file transfer protocols (SFTP).

3 Scenarios

3.1 Get Catalogue

3.1.1 Overview

The system needs to regularly receive a data file. The Subscriber requests from the Subscription Manager a list of available files within the File Distribution System to which it may subscribe.

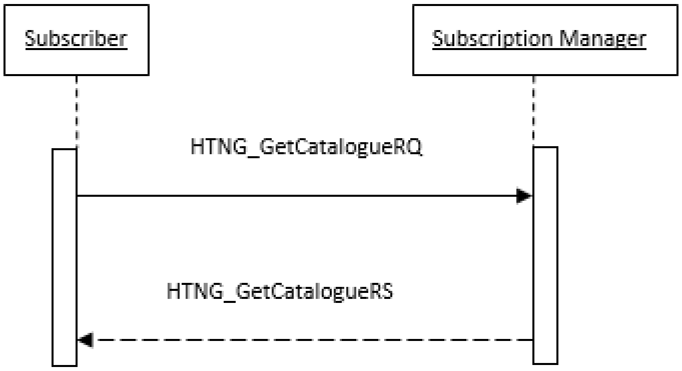
3.1.2 Roles

Role Name	Definition	Example
Subscriber	The system that subscribes to receive files	PMS systems needing to receive reservation modifications from the Central Reservation System
Subscription Manager	A web service that accepts requests to manage, gets the status of, renews and/or cancels subscriptions on behalf of an FDS	The Integration Platform

3.1.3 Use Case

Assumptions	The subscription capability exists on the File Distribution System (FDS)
Pre-conditions	The systems know about each other and have the appropriate authentication in place to communicate
Trigger	The Subscriber wishes to subscribe to receive a file, and needs a list of available files to which it may subscribe
Basic Course of Events	• The Subscriber requests a list of all available files from the Subscription Manager • The Subscription Manager responds with a list of files available within the FDS for subscription to that Subscriber, along with the URL where the subscription must be submitted • If no subscriptions are available, an empty set of data is returned
Post-conditions	None
Exception Path	None
Alternative Paths	None

3.1.4 Message Flow



3.2 Subscribe

3.2.1 Overview

A system needs to regularly receive a data file. The Subscriber provides a unique subscriber ID and a desired delivery endpoint to create a subscription through the Subscription Manager. When a file is ready to be sent, the FDS transforms the file as requested and sends the file to the URL in the subscription.

3.2.2 Roles

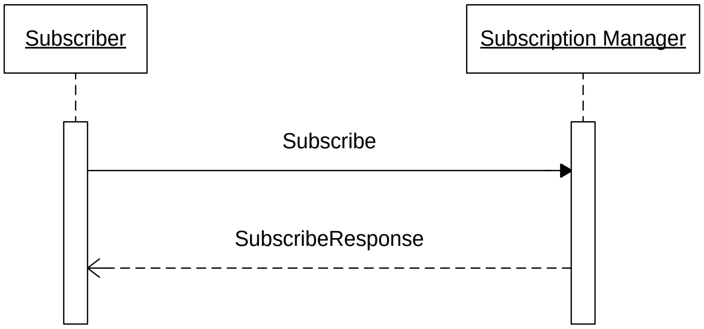
Role Name	Definition	Example
Subscriber	The system that subscribes to receive files	PMS systems needing to receive reservation modifications from the Central Reservation System
Subscription Manager	A web service that accepts requests to manage, gets the status of, renews and/or cancels subscriptions on behalf of an FDS	The Integration Platform
File Distribution System (FDS)	The system responsible for distributing files	The Central Reservation System
File Receiver	The system that will receive the files	The PMS

3.2.3 Use Case

Assumptions	The subscription capability exists on the FDS
Pre-conditions	The systems know about each other and have the appropriate authentication in place to communicate
Trigger	The Subscriber wishes to subscribe to receive a file

Basic Course of Events	• The Subscriber issues a request including a unique subscriber ID and a desired delivery endpoint to the Subscription Manager to subscribe to a particular set of files (based on what is available) • The Subscription Manager creates the subscription within the FDS • The Subscription Manager returns either message with a unique subscription ID or a fault to the Subscriber
Post-conditions	The File Receiver will receive the requested file from the FDS on the requested regular basis
Exception Path	If the subscription request is not valid, a fault will be returned (e.g. the subscription request was for a FileTypeID that does not exist) Possible Faults: • DeliveryModeRequestUnavailable • InvalidExpirationTime • UnsupportedExpirationType • FilteringNotSupported • FilteringRequestUnavailable • FileSourceUnableToProcess • InsufficientPermissions • UnknownSubscriber
Alternative Paths	None

3.2.4 Message Flows



3.3 Unsubscribe

3.3.1 Overview

A system no longer needs to regularly receive a data file. The Subscriber unsubscribes from an existing subscription in the FDS through the Subscription Manager.

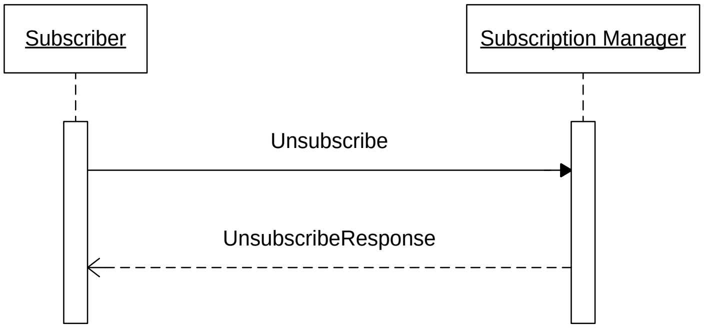
3.3.2 Roles

Role Name	Definition	Example
Subscriber	The system that subscribes to receive files	PMS systems needing to receive reservation modifications from the Central Reservation System
Subscription Manager	A web service that accepts requests to manage, gets the status of, renews and/or cancels subscriptions on behalf of an FDS	The Integration Platform
File Distribution System (FDS)	The system responsible for distributing files	The Central Reservation System
File Receiver	The system that will receive the files	The PMS

3.3.3 Use Case

Assumptions	• The subscribe/unsubscribe capability exists on the FDS • The Subscriber is currently subscribed to receive a file
Pre-conditions	The systems know about each other and have the appropriate authentication in place to communicate
Trigger	The Subscriber wishes to no longer subscribe to receive a file
Basic Course of Events	• The Subscriber sends an unsubscribe request including the appropriate subscriber ID and subscription ID to the Subscription Manager • The Subscription Manager removes the subscription in the FDS • The Subscription Manger returns the response to the Subscriber
Post-conditions	The File Receiver will no longer receive the unsubscribed files
Exception Path	If the subscription ID supplied by the Subscriber is not valid, the Subscription Manager will return an UnknownSubscription fault Possible Faults: • UnknownSubscription • InsufficientPermissions • UnknownSubscriber
Alternative Paths	None

3.3.4 Message Flows



3.4 Get Subscription Status

3.4.1 Overview

A Subscriber can determine the status of an individual subscription by requesting the subscription status from the Subscription Manager. A Subscriber can also request the status of all of its subscriptions from the Subscription Manager.

3.4.2 Roles

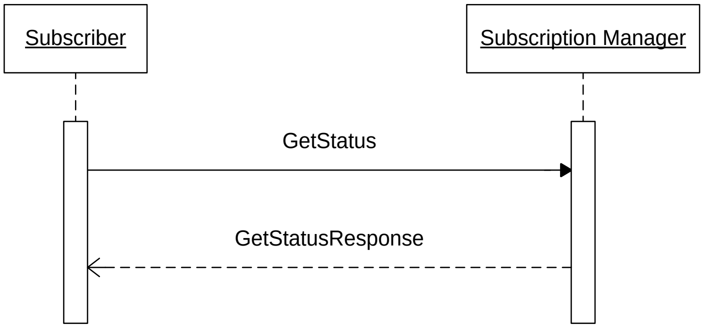
Role Name	**Definition**	**Example**
Subscriber	The system that subscribes to receive files	PMS systems needing to receive reservation modifications from the Central Reservation System
Subscription Manager	A web service that accepts requests to manage, gets the status of, renews and/or cancels subscriptions on behalf of an FDS	The Integration Platform
File Distribution System (FDS)	The system responsible for distributing files	The Central Reservation System
File Receiver	The system that will receive the files	The PMS

3.4.3 Use Case

Assumptions	• The subscription capability exists on the FDS • The Subscriber has previously subscribed to receive a file

Pre-conditions	The systems know about each other and have the appropriate authentication in place to communicate
Trigger	The Subscriber wishes to know the status of the current subscription
Basic Course of Events	• The Subscriber issues a request including the appropriate subscriber ID or subscription ID, to the Subscription Manager to determine if a subscription is active • The Subscription Manager responds with the details of the active subscription • Note: If the Subscriber issues the request using only the subscriber ID and does not specify a subscription ID, the Subscription Manager will return the details of all active subscriptions for that Subscriber • If no subscriptions are available, an empty set of data is returned
Post-conditions	None
Exception Path	If the subscription ID supplied by the Subscriber is not valid, the Subscription Manager will return a fault Possible Faults: • UnknownSubscription • UnknownSubscriber
Alternative Paths	None

3.4.4 Message Flows



3.5 Renew Subscription

3.5.1 Overview

When a system requests to regularly receive files from the File Distribution System, the subscription includes an expiration. In this use case, the Subscriber wishes to extend the subscription beyond the original expiration.

3.5.2 Roles

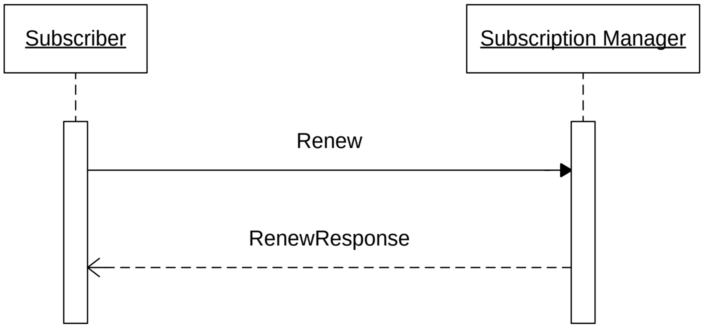
Role Name	Definition	Example
Subscriber	The system that subscribes to receive files	PMS systems needing to receive reservation modifications from the Central Reservation System
Subscription Manager	A web service that accepts requests to manage, gets the status of, renews and/or cancels subscriptions on behalf of an FDS	The Integration Platform
File Distribution System (FDS)	The system responsible for distributing files	The Central Reservation System
File Receiver	The system that will receive the files	The PMS

3.5.3 Use Case

Assumptions	• The subscription capability exists on the FDS • The Subscription Manager supports subscription renewal • The Subscriber has previously subscribed to receive a file
Pre-conditions	The systems know about each other and have the appropriate authentication in place to communicate
Trigger	The Subscriber wishes to extend a current subscription
Basic Course of Events	• The Subscriber sends a request including the appropriate subscriber ID and subscription ID, to the Subscription Manager to renew a subscription by updating the subscription expiration date • The Subscription Manager accepts the renewal request and replies with a SubscriptionID and an expiration date or a fault
Post-conditions	The subscription expiration timestamp is updated to the value passed in the renewal request

Exception Path	If the Subscription Manager cannot renew the subscription, a fault is returned to the Subscriber Possible Faults: • UnknownSubscription • InvalidExpirationTime • UnsupportedExpirationType • UnableToRenew • InsufficientPermissions • UnknownSubscriber
Alternative Paths:	None

3.5.4 Message Flows



3.6 Subscription End

3.6.1 Overview

The Subscription Manager or the FDS needs to terminate a subscription before the expiration of the subscription and notify the Subscriber of the unexpected termination.

3.6.2 Roles

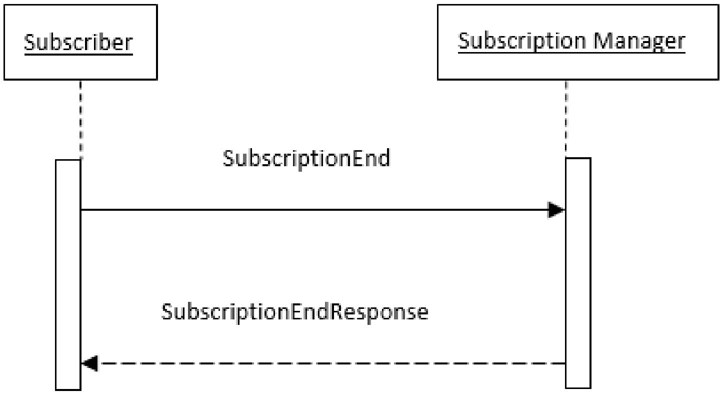
Role Name	Definition	Example
Subscriber	The system that subscribes to receive files	PMS systems needing to receive reservation modifications from the Central Reservation System
Subscription Manager	A web service that accepts requests to manage, gets the status of, renews and/or cancels subscriptions on behalf of an FDS	The Integration Platform

File Distribution System (FDS)	The system responsible for distributing files	The Central Reservation System
File Receiver	The system that will receive the files	The PMS

3.6.3 Use Case

Assumptions	The systems know about each other and have the appropriate authentication in place to communicate
Pre-conditions	The Subscriber has previously subscribed to receive a file
Trigger	The Subscription Manager or FDS needs to end a subscription early
Basic Course of Events	The Subscription Manager sends the message including the appropriate subscription ID to the Subscriber, notifying the Subscriber that the subscription is cancelled and offering a cancellation reason
Post-conditions	The subscription for the Subscriber is no longer active and the File Receiver will no longer receive the unsubscribed files
Exception Path	None
Alternative Paths	None

3.6.4 Message Flows



3.7 File Sent Notification

The Subscription Manager or the FDS has attempted to send a file and needs to update the subscriber on the status of the attempt. The attempt may have been successful or may have failed.

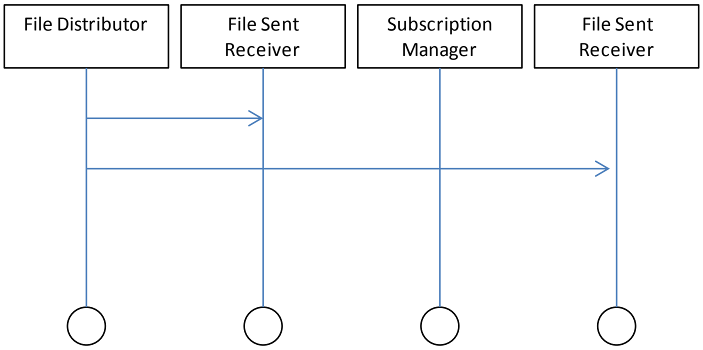
3.7.1 Roles

Role Name	Definition	Example
Subscriber	The system that subscribes to receive files	PMS systems needing to receive reservation modifications from the Central Reservation System
Subscription Manager	A web service that accepts requests to manage, gets the status of, renews and/or cancels subscriptions on behalf of an FDS	The Integration Platform
File Distribution System (FDS)	The system responsible for distributing files	The Central Reservation System
Notification Endpoint	The service endpoint that will receive the notification message	The system that maintains or requests file transfer records or attempts

3.7.2 Use Case

Assumptions	The systems know about each other and have the appropriate authentication in place to communicate
Pre-conditions	The Subscriber has previously subscribed to receive a file and provided an endpoint to receive file sent messages
Trigger	The Subscription Manager or FDS sends a file to a subscriber
Basic Course of Events	The Subscription Manager sends the message to the endpoint established by the Subscriber, notifying the subscriber that an attempt to send a file has been made
Post-conditions	The subscriber has access to the result of the attempt to send the file
Exception Path	None
Alternative Paths	None

3.7.3 Message Flows



3.8 File Ready Notification

In this scenario, the subscriber desires to retrieve files rather than having files sent. The FDS sends a notification to the end point provided with the subscription that a file is ready for retrieval.

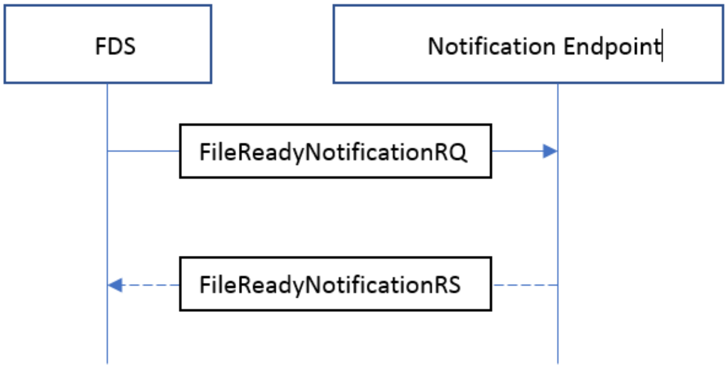
3.8.1 Roles

Role Name	**Definition**	**Example**
Subscriber	The system that subscribes to receive notifications that a file is ready to be retrieved	PMS systems needing to receive reservation modifications from the Central Reservation System
Subscription Manager	A web service that accepts requests to manage, gets the status of, renews and/or cancels subscriptions on behalf of an FDS	The Integration Platform
File Distribution System (FDS)	The system that has the files for distribution	The Central Reservation System
Notification Endpoint	The service endpoint that will receive the notification message for the subscriber	This may be the subscribing system

3.8.2 Use Case

Assumptions	The systems know about each other and have the appropriate authentication in place to communicate
Pre-conditions	The Subscriber has previously subscribed to receive a notification that a file is ready to retrieve
Trigger	The Subscription Manager or FDS has a file that is ready to be retrieved
Basic Course of Events	The FDS sends the message to the endpoint established by the Subscriber, notifying the subscriber one or more files are ready to be retrieved
Post-conditions	The subscriber is able to retrieve the files
Exception Path	None
Alternative Paths	None

3.8.3 Message Flows



4 Messages

4.1 Get Catalogue

This message is used to retrieve a collection of file descriptions available for subscription.

4.1.1 Data Element Table – Request

Element | @Attribute	Cardinality	Data Type	Description/Contents
HTNG_GetCatalogueRQ	1		The root element of the message

4.1.2 Data Element Table – Response

Element | @Attribute	Cardinality	Data Type	Description/Contents
HTNG_GetCatalogueRS	1		The root element of the message
CatalogueItems	0..1		Catalogue items
CatalogueItem	1..n		A single catalogue element
Fileset	1		Collection of files available for subscription
@PatternStyle	1	Enumeration	The file pattern style. Enumeration: Simple, Regex or Filter, default simple
Address	1	anyURI	Base URL of the desired files
FileSelector	1	string	Pattern or filter of the files to be sent
Description	1	string	Description of the available collection of files
SubscribeTo	1	anyURI	The endpoint that should be used to subscribe to these files

4.1.3 Sample Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_GetCatalogueRQ/>
    xsi:schemaLocation="http://htng.org/2017A HTNG_GetCatalogueRS.xsd"
    xmlns="http://htng.org/2017A"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
</HTNG_GetCatalogueRQ
```

4.1.4 Sample Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_GetCatalogueRS
    xsi:schemaLocation="http://htng.org/2017A HTNG_GetCatalogueRS.xsd"
    xmlns="http://htng.org/2017A"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <CatalogueItems>
    <CatalogueItem>
      <Fileset PatternStyle="Filter">
        <Address>http://www.altova.com</Address>
        <FileSelector>String</FileSelector>
      </Fileset>
      <Description>String</Description>
      <SubscribeTo>http://www.altova.com</SubscribeTo>
    </CatalogueItem>
  </CatalogueItems>
</HTNG_GetCatalogueRS>
```

4.2 Subscribe

4.2.1 Data Element Table – Request

Element | @Attribute	Cardinality	Data Type	Description/Contents
HTNG_SubscribeRQ	1		The root element of the message
Fileset	1		Collection of files that are being subscribed to
@PatternStyle	0..1	Enumeration	The file pattern style. Enumeration: Simple, Regex or Filter, default simple
@Recurse	0..1	Boolean	If true, search for files below the base directory. If false, only search specified directory.
Address	1	anyURI	Base URL of the desired files, sometimes referred to as a “top level directory”
FileSelector	1	string	Pattern or filter of the files to be sent
Delivery	1		This element contains the delivery target and may contain either the FileReadyNotification or the DeliveryNotification
FileReadyNotification	0..1	anyURI	Endpoint to receive notifications that a file is ready for download (optional). Use this when the files will be retrieved by the subscriber.
DeliveryNotification	0..1	anyURI	Endpoint to receive notifications detailing the status of an attempt to deliver a file (optional). Use this to monitor the status of attempts to send files.
SendTo	0..1	anyURI	URL where file will be sent (optional). Use this to provide a destination for where files will be sent.
Schedule	0..1		This is used to determine how frequently the system will check for files to be sent. Repeating Time Interval expressed in ISO 8601 form using -- For example: to repeat every 4 hours starting at 1:00 PM: RT13:00:00-5:00--P4H
@Randomize	1	string	A value to randomize the time scheduled +/- a period or %. This allows hotels to avoid major bottlenecks if multiple systems transmit data on the same schedule.

Interval	1	dateTime	Repeating time interval expressed in ISO8601
FileEvent	1	Enumeration	Specifies that a file will be sent after a file close event. Enumeration: Close
Format	0..1	string	The desired format for the file.
SubscriberID	0..1	string	The unique ID used by subscribing system to refer to this subscription internally. If a SubscriberID is sent in the request, it must also be sent in the response.
EndTo	0..1	anyURI	URI to receive subscription end notifications.
Expires	0..1	dateTime	Desired expiration date and time for a subscription expressed in ISO 8601 format. If absent, the Subscription Manager will determine the date.

4.2.2 Data Element Table – Response

Element |@Attribute	Cardinality	Data Type	Description/Contents
HTNG_SubscribeRS	1		The root element of the message
SubscriptionID	1	string	The ID of the subscription record to identify the subscription on the Subscription Manager
Expires	1	dateTime	When the subscription expires

4.2.3 Sample Request

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_SubscribeRQ
    xsi:schemaLocation="http://htng.org/2017A HTNG_SubscribeRQ.xsd"
    xmlns="http://htng.org/2017A"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Fileset PatternStyle="Regex">
    <Address>http://www.filesource.com</Address>
    <FileSelector>^.*\.log</FileSelector>
  </Fileset>
  <Delivery>
    <SendTo>http://www.filedestination.com</SendTo>
  </Delivery>
  <Schedule Randomize="00:10:00">
    <Interval>RT13:00:00-5:00--P4H</Interval>
  </Schedule>
  <Format>CSV</Format>
  <EndTo>http://www.endmessage.com</EndTo>
  <Expires>2001-12-17T00:00:00Z</Expires>
</HTNG_SubscribeRQ>
-17T00:00:00Z</Expires>
</HTNG_SubscribeRQ>
```

4.2.4 Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_SubscribeRS
    xsi:schemaLocation="http://htng.org/2017A HTNG_SubscribeRS.xsd"
    xmlns="http://htng.org/2017A"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SubscriptionID>1234-5678-90AB-CDEF</SubscriptionID>
  <Expires>2001-12-17T09:30:47Z</Expires>
</HTNG_SubscribeRS>
```

4.3 Unsubscribe

The Subscriber may use this message to cancel an existing subscription.

4.3.1 Data Element Table – Request

Element | @Attribute	Cardinality	Data Type	Description/Contents
HTNG_UnsubscribeRQ	1		The root element of the message
SubscriptionID	1	string	The ID of the subscription record for this subscriber

4.3.2 Data Element Table – Response

Element | @Attribute	Cardinality	Data Type	Description/Contents
HTNG_UnsubscribeRS	1		The root element of the message
SubscriptionID	0..1	string	The ID of the subscription record just cancelled, the ID no longer refers to a valid subscription

4.3.3 Sample Request

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_UnsubscribeRS
    xsi:schemaLocation="http://htng.org/2017A HTNG_UnsubscribeRS.xsd"
    xmlns="http://htng.org/2017A"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SubscriptionID>1234-5678-90AB-CDEF</SubscriptionID>
</HTNG_UnsubscribeRS>
```

4.3.4 Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_UnsubscribeRS
    xsi:schemaLocation="http://htng.org/2017A HTNG_UnsubscribeRS.xsd"
    xmlns="http://htng.org/2017A"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SubscriptionID>1234-5678-90AB-CDEF</SubscriptionID>
</HTNG_UnsubscribeRS>
```

4.4 Get Subscription Status

This message is used to determine if a subscription is active; it returns the current expiration date of the subscription.

4.4.1 Data Element Table - Request

Element | @Attribute	Cardinality	Data Type	Description/Contents
HTNG_GetSubscriptionStatusRQ	1		The root element of the message
SubscriptionID	1	string	The ID of the subscription record for this subscriber

4.4.2 Data Element Table - Response

Element | @Attribute	Cardinality	Data Type	Description/Contents
HTNG_GetSubscriptionStatusRS	1		The root element of the message
SubscriptionID	1	string	The ID of the subscription record to identify the subscription on the subscription manager
Expires	1	dateTime	When the subscription expires

4.4.3 Sample Request

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_GetSubscriptionStatusRQ
    xsi:schemaLocation="http://htng.org/2017A HTNG_GetSubscriptionStatusRQ.xsd"
    xmlns="http://htng.org/2017A"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SubscriptionID>1234-5678-90AB-CDEF</SubscriptionID>
</HTNG_GetSubscriptionStatusRQ>
```

4.4.4 Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_GetSubscriptionStatusRS
    xsi:schemaLocation="http://htng.org/2017A HTNG_GetSubscriptionStatusRS.xsd"
    xmlns="http://htng.org/2017A"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SubscriptionID>1234-5678-90AB-CDEF</SubscriptionID>
  <Expires>2001-12-17T09:30:47Z</Expires>
</HTNG_GetSubscriptionStatusRS>
```

4.5 Renew Subscription

This message is used to extend a subscription before it expires.

4.5.1 Data Element Table - Request

Element | @Attribute	Cardinality	Data Type	Description/Contents
HTNG_RenewSubscriptionRQ	1		The root element of the message

SubscriptionID	1	string	The ID of the subscription record for this subscriber
Expires	1	dateTime	Modified or extended subscription expiration date

4.5.2 Data Element Table - Response

Element | @Attribute	Cardinality	Data Type	Description/Contents
HTNG_RenewSubscriptionRS	1		The root element of the message
SubscriptionID	1	string	The ID of the subscription record for this subscriber
Expires	1	dateTime	When the subscription expires

4.5.3 Sample Request

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_RenewSubscriptionRQ
    xsi:schemaLocation="http://htng.org/2017A HTNG_RenewSubscriptionRQ.xsd"
    xmlns="http://htng.org/2017A"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SubscriptionID>1234-5678-90AB-CDEF</SubscriptionID>
  <Expires>2001-12-17T09:30:47Z</Expires>
</HTNG_RenewSubscriptionRQ>
```

4.5.4 Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_RenewSubscriptionRS
    xsi:schemaLocation="http://htng.org/2017A HTNG_RenewSubscriptionRS.xsd"
    xmlns="http://htng.org/2017A"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SubscriptionID>1234-5678-90AB-CDEF</SubscriptionID>
  <Expires>2001-12-17T09:30:47Z</Expires>
</HTNG_RenewSubscriptionRS>
```

4.5.5 Expiration Note

If the requested expiration date is beyond the latest acceptable date, the system will respond by extending the subscription to the latest acceptable date.

4.6 Subscription End

This notification message is used by the Subscription Manager to inform the Subscriber the subscription is being cancelled earlier than planned.

4.6.1 Data Element Table - Request

Element | @Attribute	Cardinality	Data Type	Description/Contents
HTNG_SubscriptionEndRQ	1		The root element of the message
SubscriptionID	1	string	The ID of the subscription record that is being cancelled

Status	1	string	The status of this subscription is either: SourceShuttingDown, SourceCancelling or DeliveryFailure
Reason	0..1	string	The reason of the cancellation for the subscription

4.6.2 Data Element Table – Response

Element | @Attribute	Cardinality	Data Type	Description/Contents
HTNG_SubscriptionEndRS	1		The root element of the message
SubscriptionID	1	string	The ID of the subscription record that is being cancelled

4.6.3 Sample Request

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_SubscriptionEndRQ
    xsi:schemaLocation="http://htng.org/2017A HTNG_SubscriptionEndRQ.xsd"
    xmlns="http://htng.org/2017A"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SubscriptionID>1234-5678-90AB-CDEF</SubscriptionID>
    <Status>SourceCancelling</Status>
    <Reason>Source shutting down</Reason>
</HTNG_SubscriptionEndRQ>
```

4.6.4 Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_SubscriptionEndRS
    xsi:schemaLocation="http://htng.org/2017A HTNG_SubscriptionEndRS.xsd"
    xmlns="http://htng.org/2017A"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SubscriptionID>1234-5678-90AB-CDEF</SubscriptionID>
</HTNG_SubscriptionEndRS>
```

4.7 File Sent Notification

The notification message is sent by the FDS to inform the subscribing system that an attempt to send a file has been made. The message includes information about the success or failure of the attempt.

4.7.1 Data Element Table – Request

Element | @Attribute	Cardinality	Data Type	Description/Contents
HTNG_FileSentNotificationRQ	1	element	The root element of the message
-SubscriptionID	1	string	The ID of the subscription record for this subscriber
-Filename	1	string	Name of the file sent
-FileDateTime	1	Date Time	Date and time of last file modification

–FileSize	1	integer	Size of file in bytes
–FilePath	1	string	Path to file relative to some configured root
–CheckValue	0..3	hex string	Calculated file check value
–@CheckAlgorithm	0..1	enumerated list	One of MD5, SHA1, SHA256
–AttemptDateTime	1	Date Time	Date and time of the file delivery attempt
–RetryCount	1	Integer	Number of retries after first attempt
–Result	1	Element	Describes the result of an attempt to send a file
––SuccessCode	1	Integer	0 if successful, else integer failure code
––Reason	0..1	String	Brief text explanation of code

4.7.2 Data Element Table – Response

This is an optional response to a FileSentNotification.

Element | @Attribute	Cardinality	Description/Contents
HTNG_FileSentNofitificationRS	1	The root element of the message
–SubscriptionID	1	The ID of the subscription record for this subscriber

4.7.3 Sample Request

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_FileSentNotifRQ
    xsi:schemaLocation="http://htng.org/2017A HTNG_FileSentNotifRQ.xsd"
    xmlns="http://htng.org/2017A"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SubscriptionID>1234-5678-90AB-CDEF</SubscriptionID>
  <FileName>DailyCYBWI.log</FileName>
  <FileDateTime>2001-12-17T09:30:47Z</FileDateTime>
  <FileSize>2048</FileSize>
  <FilePath>/server-CYBWI/mothership/daily</FilePath>
  <CheckValue CheckAlgorithm="MD5">7e9063829a73b64c8a0045c3274c1857</CheckValue>
  <AttempDateTime>2001-12-17T09:30:47Z</AttempDateTime>
  <RetryCount>0</RetryCount>
  <Result>
    <SucessCode>0</SucessCode>
    <Reason>Success</Reason>
  </Result>
</HTNG_FileSentNotifRQ>
```

4.7.4 Sample Response

This is an optional response:

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_FileSentNotifRS
    xsi:schemaLocation="http://htng.org/2017A HTNG_FileSentNotifRS.xsd"
    xmlns="http://htng.org/2017A"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SubscriptionID>1234-5678-90AB-CDEF</SubscriptionID>
</HTNG_FileSentNotifRS>
```

4.8 File Ready Notification

This message is sent by the FDS to the end point established by the subscriber when one or more files are ready to be retrieved by the subscriber.

4.8.1 Data Element Table - Request

Element | @Attribute	Cardinality	Data Type	Description/Contents
HTNG_FileReadyNotificationRQ	1	element	The root element of the message
-SubscriptionID	1	string	The ID of the subscription record for this subscriber
-Files	1	Collection	Collection of files to retrieve
-./File	1..N	element	
--./FileURI	1	element	URI of files to retrieve
--./CheckValue	0..3	hex string	Calculated file check value
-@CheckAlgorithm	0..1	enumerated list	One of MD5, SHA1, SHA 256

4.8.2 Data Element Table - Response

This is an optional response to the FileReady Notification.

Element | @Attribute	Cardinality	Description/Contents
HTNG_FileReadyNotificationRS	1	The root element of the message
-SubscriptionID	1	The ID of the subscription record for this subscriber

4.8.3 Sample Request

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_FileReadyNotifRQ
    xsi:schemaLocation="http://htng.org/2017A HTNG_FileReadyNotifRQ.xsd"
    xmlns="http://htng.org/2017A"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SubscriptionID>1234-5678-90AB-CDEF</SubscriptionID>
  <Files>
    <File>
      <FileURI>http://www.altova.com</FileURI>
      <CheckValue CheckAlgorithm="MD5">7e9063829a73b64c8a0045c3274c1857</CheckValue>
    </File>
  </Files>
</HTNG_FileReadyNotifRQ>
```

4.8.4 Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<HTNG_FileReadyNotifRS
    xsi:schemaLocation="http://htng.org/2017A HTNG_FileReadyNotifRS.xsd"
    xmlns="http://htng.org/2017A"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SubscriptionID>1234-5678-90AB-CDEF</SubscriptionID>
</HTNG_FileReadyNotifRS>
```

5 Simple Filters for Bulk Data

A simple filtering mechanism for controlling the selection of files to be sent in a bulk transfer subscription may be valuable to some implementers. This mechanism was created for use in the HTNG Event Notification 3.0 Specification and has been proposed as a basis for use in Bulk File subscriptions. In the Event Notification 3.0 Specification, the filter is applied to the content of the message generated by the event. In the Bulk File case, the filter is based on the file meta-data attributes.

5.1 File Meta-Data

The elements of the common file meta-data object are as follows:

- Filename – the simple name of the file (case sensitive)
- Creation Time – the time the file was created in ISO 8601 format
- Modification Time – the time the file last modified in ISO 8601 format
- Mime Type – multipurpose Internet mail extension
- Subfolder Name – the path of any subfolder below the subscribed root, with support for wildcards
- File Size – the size of the file in bytes
- Age – the age of the file expressed as an ISO 8601 duration

Additional elements may be provided, but are implementation dependent. In addition, special values are supported:

- $Today – today's date
- $Now – the current date and time
- $Dow – the current day of the week expressed as a number with 1=Monday
- $Year – the current 4-digit year
- $Month – the current month as an integer value from 1 to 12 with 1=January
- $Day – the current day of the month as an integer
- $Hour – the current hour of the day as an integer from 0-24
- $Minute – the current minute of the hour as an integer from 0-59
- $Second – the current second of the minute

5.2 Simple Filter Schema

5.2.1 Data Element Table – Request

The schema for the HTNG_SimpleFilter is:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.htng.org/htngSimpleFilter"
  xmlns="http://www.htng.org/htngSimpleFilter"
  elementFormDefault="qualified">
  <!-- defines a type for the name element with two attributes rule and type -->
  <xsd:complexType name="NameType">
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute
          name="rule"
          type="xsd:string" />
        <xsd:attribute
          name="type"
          type="xsd:string" />
```

5.2.2 Data Element Table – Response

```
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>

  <!--
    MatchType defines a type for match elements.
    A match element may have either a name-value pair or
    a list of match elements
  -->
  <xsd:complexType name="MatchType">
    <xsd:choice>
      <xsd:sequence>
        <xsd:element
          name="name"
          type="NameType" />
        <xsd:element
          name="value"
          type="xsd:string" />
      </xsd:sequence>
      <xsd:sequence>
        <xsd:choice maxOccurs="unbounded">
          <xsd:element
            name="matchAny"
            type="MatchType" />
          <xsd:element
            name="matchAll"
            type="MatchType" />
          <xsd:element
            name="matchOne"
            type="MatchType" />
          <xsd:element
            name="matchNone"
            type="MatchType" />
        </xsd:choice>
      </xsd:sequence>
    </xsd:choice>
  </xsd:complexType>

  <!-- An HTNG_SimpleFilter must have a single element of any matchType element -->
  <xsd:complexType name="HTNG_SimpleFilterType">
    <xsd:choice>
      <xsd:element
        name="matchAny"
        type="MatchType" />
      <xsd:element
        name="matchAll"
        type="MatchType" />
      <xsd:element
        name="matchOne"
        type="MatchType" />
      <xsd:element
        name="matchNone"
        type="MatchType" />
    </xsd:choice>
  </xsd:complexType>

  <xsd:element
    name="HTNG_SimpleFilter"
    type="HTNG_SimpleFilterType" />

</xsd:schema>
```

5.3 Element HTNG_SimpleFilter

The HTNG_SimpleFilter element is the root element for all simple filters. The filter contains a single matchAll, matchAny, matchOne, or matchNone element. A file is sent if the match element matches the files meta-data. If the filter does not match the meta-data, the file will be blocked and not sent.

5.4 Element matchAll

A matchAll element matches if all of the elements it contains match their contents. The element can contain a mixed list of matchAll, matchAny, matchNone or matchOne elements or a single name element and a list of value elements. A matchAll element may complete processing immediately after it encounters a failed match.

5.5 Element matchAny

A matchAny element matches if any one (or more) of the elements it contains match their content. The element can contain a mixed list of matchAll, matchAny, matchOne or matchNone elements or a single name element and a list of value elements. A matchAny element may complete processing immediately upon the first successful match.

5.6 Element matchNone

A matchNone element matches if none of the elements it contains match their content. The element can contain a mixed list of matchAll, matchAny or matchNone elements or a single name element and a list of value elements. When this element contains a single matchAll or matchAny element, it essentially negates the result of the contained match. A matchNone element may complete processing immediately upon the first successful match.

5.7 Element matchOne

A matchOne element matches only if exactly one of the elements it contains matches its contents. The element can contain a mixed list of matchAll, matchAny, matchOne or matchNone elements, or a single name element and a list of value elements. A matchOne element may complete processing immediately upon the second successful match.

5.8 Element Name

The primary difference between the name element of this specification and that of the Event Notification Specification 3.0, is that the name element must be one of the file meta-data elements.

Only one name element is allowed in a match element. The name element contains a simple character string that identifies or references a value to be tested or compared. If a value element does not follow a name element, the message will succeed if the name exists within the evaluation context. How the name is interpreted as a reference or how the name is processed is left up to the implementation. If the name does not exist or is not valid within the evaluation context, the subscription request with the filter should be rejected by the source.

The name element has two optional attributes: rule and type. The rule attribute identifies how the name and value will be compared or matched. If the rule attribute is not provided or supported, the match will be a regular expression match using the values as regular expressions.

The type attribute identifies the data type for interpreting the value elements. If the type attribute is not provided then the type will be assumed to match the default type of the named element.

5.8.1 Attribute Rule

The rule attribute is treated as a reference to Boolean comparison operation with true corresponding to "match" and false corresponding to "does not match." This is identified by using the information referenced by the name element as one argument and the value element as the second argument. If multiple value elements exist then the rule is applied once for each value until the conditions of the match are met or all values are exhausted.

Supported rules may vary by type but the following are supported by all types:

- isGreater – the named item is greater (or older) than the value
- isLess – the named item is less than (or newer than) the value
- isGreaterOrEqual – the named item is greater than or equal to the value
- isLessOrEqual – the named item is less than or equal to the value
- isEqual – the named item is equal to the value
- isNotEqual – the named item does not equal the value
- regex – the value is treated as a regular expression and applied against the named element which is treated as a string
- simple – allows for simple wildcard string matching using '?' for any single character and '*' to match a run of characters.

If times provided as values have no dates then the date is assumed to be the current date in the local time zone of the server. All other rules are optional; for example, since ISO 8601 has the ability to define time intervals an isBetween, the rule may optionally be supported.

5.8.2 Attribute Type

The optional type attribute of the name element describes the data type of the item referenced by the name element. The value elements should all be evaluated as this same type. In the absence of a type, the named items and all values will be treated as strings. The following types should be recognized:

- boolean – true or false

- double – any number with a decimal point
- date – a date or date-time
- integer – an integer
- string – a string
- time – a time
- duration – a duration of time
- interval – a time interval

Dates, times, durations, and intervals should all be expressed in ISO8601 standard formats.

5.9 Element Value

A MatchType (matchAll, matchAny, matchNone, matchNone) element may contain zero or more value elements. The value element holds a value to be compared to the item referenced by the name element. By default, the value will be treated as a regular expression and the match is made if the regular expression is matched against the named item. If multiple values are provided then each element is evaluated according to the rules of its matching container.

If the name element has a rule and a type attribute, the value should be evaluated as a value of the same type. If the name element does not have a type attribute, the value should be treated as a string. Each value should be checked as being a valid instance in its type and context prior to accepting the filter in a subscribe message.

5.10 Filter Examples

The following sections provide some examples demonstrating how the HTNG_SimpleFilters might be applied.

5.10.1 *Send only files created in the last hour*

This filter will only send files that were created in the last hour;

```
<HTNG_SimpleFilter>
  <matchAny>
     <name rule="isLessOrEqual">age</name>
     <value>P60M</value>
  </matchAny>
</HTNG_SimpleFilter>
```

5.10.2 *Match files beginning with a specific hotel code*

This filter will allow the sending of any file beginning with CYA.

```
<HTNG_SimpleFilter>
  <matchAny>
    <name rule="regex">filename</name>
    <value>^CYA.*</value>
  </matchAny>
</HTNG_SimpleFilter>
```

5.10.3 Send files modified today

This example will allow files modified today to be sent.

```
<HTNG_SimpleFilter>
  <matchAny>
    <name rule="isLessThan">modifiedTime</name>
    <value>T00:00:00-5:00</value>
  </matchAny>
</HTNG_SimpleFilter>
```

5.10.4 Match all files created in the previous three days (not today)

Using an ISO 8601 interval type for the value element allows the creation of a rule like isBetween which matches any date or time within the provided time interval. This example matches any date between the start of the day three days ago and the start of the day today.

```
<HTNG_SimpleFilter>
    <matchAny>
       <name rule="isBetween" type="interval">creationTime</name>
       <value>P3D--$Today</value>
    </matchAny>
</HTNG_SimpleFilter>
```

5.10.5 Match all files created exactly three days ago

Using two intervals allows the selection of any single day in the past. The following example matches files that are three days old, but not files for the previous two days.

```
<HTNG_SimpleFilter>
    <matchAll>
        <matchAll>
           <name rule="isBetween" type="interval">creationTime</name>
           <value>P3D--$Today</value>
        </matchAll>
        <matchNone>
           <name rule="isBetween" type="interval">creationTime</name>
           <value>P2D--$Today</value>
        </matchNone>
    <matchAll>
</HTNG_SimpleFilter>
```

6 Appendices

6.1 Glossary of Terms

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

Term	Definition
Subscriber	A system that sends requests to create, renew and/or cancel subscriptions on behalf of a File Receiver
Subscription Manager	A web service that accepts requests to manage, get the status of, renew and/or cancel subscriptions on behalf of an FDS
File Distribution System (FDS)	A system that is responsible for producing, transforming, and distributing files of data to subscribed systems
File Receiver	A system which wishes to receive files from an FDS on a regular basis

6.2 Implementation Notes

Some operating systems may not support case-sensitive searches. Depending on your regex library, a regex command may be able to be used to ignore case-sensitive search types.