



# Omneon MediaBridge™

## Web Services Description

### Copyright and Trademarks

Copyright © 1998-2011 Harmonic, Inc. Contents of this publication may not be reproduced in any form without the written permission of Harmonic, Inc. Omneon and all Omneon logos are trademarks of Harmonic, Inc.

\*All other trademarks and registered trademarks are the property of their respective holders.

### Notice

Information contained in this publication is subject to change without notice or obligation. While every effort has been made to ensure that the information is accurate as of the publication date, Harmonic, Inc. assumes no liability for errors or omissions. In addition, Harmonic, Inc. assumes no responsibility for damages resulting from the use of this publication.

---

## Table of Contents

<b>Monitoring Web Service .....</b>	<b>3</b>
GetOperations() .....	3
GetOperationDescription (string operationId) .....	3
GetStatus (string operationId) .....	3
purge_complete_and_cancelled () .....	4
purge_all_failed () .....	4
cancel_operation (string operationID) .....	4
retry_operation (string operationID) .....	4
redo_operation (string operationID) .....	4
cancel_all () .....	4
<b>Configuration Web Service .....</b>	<b>4</b>
get_registered_processing_chain_classes () .....	4
get_active_Pcs () .....	4
add_processing_chain (string processingChainClass, string xmlString) .....	5
remove_processing_chain (string processing_chain_id) .....	5
new_operation (string processing_chain_id, string xmlString) .....	5

## Monitoring Web Service

### GetOperations()

Returns all the identifiers (ID's) of all operations into an array of strings. Each string represents an identifier (ID) of one operation.

### GetOperationDescription (string operationId)

Returns the description for an operation with the given operation ID.

*Example of the returned description:*

```
<OpDescription>
  <readfromsourcepackage>False</readfromsourcepackage>
  <clip_name />
  <com_server_name>MOG.Toboggan.MediaTransfer.IPIngest</com_server_name>
  <files_to_ingest>
    <files_to_ingestItem>file:\C:\temp\input_folder_mxf\C0001.MXF</files_to_ingestItem>
  </files_to_ingest>
  <input_file>C:\temp\temp_folder\0b5dc306-ea96-4c61-bd78-763013e0375f.xml</input_file>
</OpDescription>
```

### GetStatus (string operationId)

Returns a string in the form of XML with the status of all the operations.

*Example of the returned description:*

```
<StatusList>
  <StatusItem>
    <Status>2</Status>
    <Percentage>0.0</Percentage>
    <ID>11945175323602eae7251a7f5099f4cf61fee9b048</ID>
  </StatusItem>
  <StatusItem>
    <Status>2</Status>
    <Percentage>0.0</Percentage>
    <ID>11945175413eebdcea4b0c9bff8f7434028293afa3</ID>
  </StatusItem>
</StatusList>
```

The field **Status** translation is:

- 0 - STATE\_IDLE
- 1 - STATE\_RUNNING
- 2 - STATE\_FAILED
- 3 - STATE\_PAUSED

4 - STATE\_CANCELLED  
5 - STATE\_COMPLETE  
6 - STATE\_PENDING  
7 - STATE\_CANCELLED  
8 - STATE\_CONSOLIDATING  
9 - STATE\_INITIALIZING

### **purge\_complete\_and\_cancelled ()**

Purge all of the complete and cancelled operations. Returns a boolean value which indicates if the request was successfully received or not.

### **purge\_all\_failed ()**

Cancel all failed operations. Returns a boolean value which indicates if the request was successfully received or not.

### **cancel\_operation (string operationID)**

Cancel the operation with the given operation ID. Returns a boolean value which indicates if the request was successfully received or not.

### **retry\_operation (string operationID)**

Do a “retry request” of the failed operation with the given ID. Returns a boolean value which indicates if the request was successfully received or not.

### **redo\_operation (string operationID)**

Do a “redo request” of the operation with the given ID. Returns a boolean value which indicates if the request was successfully received or not.

### **cancel\_all ()**

Cancel all operations. Returns a boolean value which indicates if the request was successfully received or not.

---

## **Configuration Web Service**

### **get\_registered\_processing\_chain\_classes ()**

Returns an array of strings with the names of the supported processing chain classes.

### **get\_active\_Pcs ()**

Returns a string with an XML describing the active processing chains.

*Return example:*

```
<active_pcs><pc><class>mog.toboggan.core.default_com_operation.CODProcessorChain</class><id>de4e77c3-f5a8-443d-a1bf-4c113e143f42</id><name>default</name></pc></active_pcs>
```

**add\_processing\_chain (string processingChainClass, string xmlString)**

Adds a new processing chain of a specific processingChainClass and returns empty string "" when succeeded. The parameter **xmlString** is the specific processing chain.

*xmlString parameter example:*

```
<processing_chain given_name="default">
  <private_data>
    <file_complete_period value="0.5"/>
    <cod_operator_id value="MOG.Toboggan.EDLDirectWrite"/>
    <cod_input_folder value="C:" />
  </private_data>
</processing_chain>
```

**remove\_processing\_chain (string processing\_chain\_id)**

Returns the list of processing chains in a string which represents an XML.

*Response example:*

```
<processing_chain max_retries="0" id="de4e77c3-f5a8-443d-a1bf-4c113e143f42"
module="mog.toboggan.core.default_com_operation.CODProcessorChain" given_name="default">
  <private_data>
    <file_complete_period value="0.5"/>
    <cod_operator_id value="MOG.Toboggan.EDLDirectWrite"/>
    <cod_input_folder value="C:"/>
  </private_data>
</processing_chain>
```

**new\_operation (string processing\_chain\_id, string xmlString)**

Creates an operation on a specific processing chain. The parameter **xmlString** is a COD.