

MARQUIS PARTIAL FILE RESTORE

INDEX WEB SERVICE API: 31ST AUGUST 2018

CONTENTS

Introduction	2
Release History	2
PFR API Calls.....	3
Indexing a File	3
Requesting File Index Status	4
Returned XML	4
Requesting File Byte Offsets	6
Returned XML	7
Partial File Request	8
Returned XML	9
Partial File Request Status	10
Returned XML	10

INTRODUCTION

RELEASE HISTORY

DOCUMENTATION RELEASE 30TH AUGUST 2018

Major new release.

Full API update to support new PFR architecture with flexible source locations, centralised Index storage and flexible output file location control.

NOTE: All paths MUST be full UNC and must be accessible by the user supplied to the service configuration

PFR API CALLS

The PFR Index Web Service API is a REST Interface that listens on TCP/IP port 60792.

INDEXING A FILE

Call Name	indexfile
Call Type	GET
Parameter	<p>filepath</p> <p>The full UNC path to the media file to be indexed.</p> <p>e.g. filepath=\\server\\share\\folder\\movie_to_be_indexed.mxf</p> <p>indexid</p> <p>The unique identifier for the source file. This will be used as the reference for this file throughout the following.</p> <p>e.g. indexid=unique_text_id</p>
Returns	See section below – Returned XML (shared with filestatus call)

NOTES

This method will block while the index is created and will only return when either the index file has been created or for some reason it has not been possible to create the index file.

The Web Service will support multiple concurrent calls to this command.

EXAMPLE

The following example shows the HTTP request to send PFRIndexer a request to index the media file \\servername\\share\\File.mxf

<http://PFRIndexer:60792/indexfile?filepath=\\server\\share\\File1.mxf&indexid=uniqueid1>

REQUESTING FILE INDEX STATUS

Call Name	filestatus
Call Type	GET
Parameter	indexid The unique identifier for the source file. e.g. indexid=unique_text_id
Returns	See section below – Returned XML

NOTES

This method will block while retrieving the index status for a previously indexed file.

This method internally uses an XML file, generated by the indexer to retrieve the detailed status.

The Web Service will support multiple concurrent calls to this API call.

EXAMPLE

The following example shows the HTTP request to request from server PFRIndexer the status of media file \\servername\share\File1.mov:

<http://PFRIndexer:60792/filestatus?indexid=uniqueid1>

RETURNED XML

The return from either of the above calls is a packet of XML that describes the result of the call. In effect, internally, when requested to perform an index on a file, we then request its status and make that the return packet of the call.

Unindexed File

The filestatus message when a file not indexed

```
<?xml version="1.0" encoding="utf-8" ?>
<IndexerReport IndexResult="Not Indexed"/>
```

Indexing

The message when the indexing is in progress

```
<?xml version="1.0" encoding="UTF-8" ?>
<IndexerReport IndexResult="Indexing"/>
```

Failed To Index

The message and error information when a file has failed to index.

```
<?xml version="1.0" encoding="utf-8" ?>
<IndexerReport IndexResult="Failed" IndexTime="2011/10/21 15:30:15"
errorCode="errornumber" errorStr="error description"/>
```

Succeeded

The result of a good fileindex or filestatus call

```
<?xml version="1.0" encoding="UTF-8" ?>

<IndexerReport IndexResult="Succeeded" IndexTime="2018/08/29 16:36:20"
FileStartTC="01:00:00;00" FileDuration="1800" FileFrameRate="29.97"
originalFile="//Seerver/share/File.mxf" indexID="unique_text_id1"/>
```

Call Name	fileoffsets
Call Type	GET
Parameter	<p>indexid</p> <p>The unique identifier for the source file. e.g. indexid=unique_text_id</p> <p>tcin</p> <p>Timecode¹ of the first frame requested</p> <p>tcout</p> <p>Timecode of the last frame requested</p> <p>fileframerate</p> <p>Frame rate, as returned in the file status report</p>
Returns	See section below – Returned XML

NOTES

This method will block whilst retrieving the start and end byte offsets for the requested timecodes. The offsets are extended in order to handle GOP and interleave ordering.

If the tcout requests the last frame or beyond from the source the return will give ‘0xffffffff’.

EXAMPLE

The following example shows the HTTP request to retrieve the byte offsets from the media file \\servername\File1.mov:

```
http://PFRIndexer:60792/fileoffsets?indexid=unique_text_id1&tcin=00:00:10:00&tcout=00:05:00:00&fileframerate=25
```

¹ Timecode format should be in form hh:mm:ss:ff for non-drop framerates and hh:mm:ss;ff for drop framerates

RETURNED XML

The return from the **fileoffsets** call is a packet of XML that describes the result of the call.

Good fileoffsets call

```
<?xml version="1.0" encoding="utf-8" ?>
< fileoffsetvalues fileoffsetsResult="Succeeded" in_bytes="0x0060000"
out_bytes="0x0080000"/>
```

fileoffsets when file not present

```
<?xml version="1.0" encoding="utf-8" ?>
< fileoffsetvalues fileoffsetsResult="Error File Not Found"/>
```

Good fileoffsets call when asking for last frame or later from the source

```
<?xml version="1.0" encoding="utf-8" ?>
<fileoffsetvalues fileoffsetsResult="Succeeded" in_bytes="0x7c28014"
out_bytes="0xffffffffffffffff"/>
```

Call Name	partialfile
Call Type	GET
Parameter	<p>indexid</p> <p>The unique identifier for the source file.</p> <p>e.g. indexid=unique_text_id</p> <p>tcin</p> <p>Timecode of the first frame requested</p> <p>tcout</p> <p>Timecode of the last frame requested</p> <p>fileframerate</p> <p>Frame rate, as returned in the file status report</p> <p>part_file</p> <p>Full UNC path to partial restored file fragment</p> <p>out_filename</p> <p>Output file name and unc path for the new partial media file.</p> <p>e.g. \\server\share\newpartialfile.mxf</p>
Returns	See section below – Returned XML

EXAMPLE

http://PFRIndexer:60792/partialfile?indexid=unique_text_id1&tcin=00:01:00:00&tcout=00:02:00:00&fileframerate=25&part_file=\\host\media\temp\partfile&out_filename=\\server\share\MyPartialRestore.mxf

NOTES

This method will use the parameters supplied to generate a Marquis XML file that will be used to create the partial output file.

Call type now GET, although PUT and POST will be accepted.

NOTE: If the output file already exists the transfer will fail.

RETURNED XML

The return from the **partialfile** call is a packet of XML that describes the result of the call.

Successful partialfile call. XML passed to the Medway file processing

```
<?xml version="1.0" encoding="UTF-8" ?>
<partialfile partialfileResult="Succeeded"/>
```

Duplicate Parameter in call

```
<?xml version="1.0" encoding="UTF-8" ?>
<partialfile partialfileResult="Error Duplicate parameter"/>
```

Missing Parameter in call

```
<?xml version="1.0" encoding="UTF-8" ?>
<partialfile partialfileResult="Error Missing parameter"/>
```

Incorrect Framerate in call

```
<?xml version="1.0" encoding="UTF-8" ?>
<partialfile partialfileResult="Error Bad framerate"/>
```

PARTIAL FILE REQUEST STATUS

Call Name	partialfilestatus
Call Type	GET
Parameter	targetpartialname The UNC path to the new partial movie file. e.g. \\server\share\newpartialfile.mxf
Returns	See section below – Returned XML

Notes

This method will return status (% complete) for the creation of a partial media file initiated using the Partial File Request API call.

RETURNED XML

For Existing Job:

```
<?xml version="1.0" encoding="utf-8" ?>  
<partialfilestatus phase="[Pending|Parsing|Transferring|Complete|Failed]"  
percentcomplete="26" errorCode="errornumber" errorStr="error description"/>
```

e.g.

```
<?xml version="1.0" encoding="UTF-8" ?>  
<partialfilestatus phase="Failed" percentcomplete="0" errorCode="-2132778983"  
errorStr="Failed to create file."/>
```

For Non-existent Job:

```
<?xml version="1.0" encoding="utf-8" ?>  
<partialfilestatus error="Job not found"/>
```