MARQUIS PARTIAL FILE RESTORE

INDEX WEB SERVICE API

CONTENTS

INTRODUCTION

NEW IN THIS VERSION

MINOR RELEASE 3RD JANUARY 2017

Error code and message added to failure responses for ***indexfile*** and ***partialfilestatus*** calls.

MINOR RELEASE 16TH DECEMBER:

Page 8 - partialfilerequest – out_filename shoud not have an extension

Page 10 – partialfilestatus – targetpartialname is case sensitive

RELEASE 15TH DECEMBER 2016:

This release include the extra API calls and functionality required for the Spectra Logic Black Pearl Partial File Restore solution and changes to the general processing.

API Call Changes:-

- partialfile
  - Updated returned XML's
  - Operates as a GET
- partialfilestatus – Request the status of the partial file creation process

General Changes:-

- Call returns updated for consistency
  - HTTP 400 errors returned for problems with the request that prevent it from being processed
  - HTTP 200 OK responses given when the request has been process even if that processing has failed

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	**filepath** The full path (via a mapped drive on the StorNext MDC) to the media file to be indexed.
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.

On file systems that leave stub files on disk, such as StorNext, if a request is made to create an index for a media file which has been truncated, this call will cause the entire media file to be restored by StorNext.

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 S:\Test\File1.mov:

```
http://PFRIndexer:60792/indexfile?filepath=S:\Test\File1.mov
```

REQUESTING FILE INDEX STATUS

Call Name	**filestatus**
Call Type	GET
Parameter	**filepath** The full path (via a mapped drive on the StorNext MDC) to the media file whose status is to be retrieved.
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 S:\Test\File1.mov:

```
http://PFRIndexer:60792/filestatus?filepath=S:\Test\File1.mov
```

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.

Failed To Index

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

Good fileindex or filestatus call

```
<?xml version="1.0" encoding="utf-8" ?>
<IndexerReport IndexResult="Succeeded" IndexTime="2011/10/21 11:40:53"
FileStartTC="01:00:00;00" FileDuration="1800" FileFrameRate="29.97"/>
```

filestatus when file not present

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

filestatus when file not indexed

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

REQUESTING FILE BYTE OFFSETS

Call Name	**fileoffsets**
Call Type	GET
Parameter	**filepath** The full path (via a mapped drive) to the media file whose partial offsets are being requested. **tcin** Timecode[1] of the first frame requested **tcout** Timecode of the last frame requested **fileframerate** Frame rate, as returned in the file status report
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.

EXAMPLE

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

```
        http://PFRIndexer:60792/fileoffsets?filepath=S:\Test\File1.mov&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"/>
```

PARTIAL FILE REQUEST

Call Name	**partialfile**
Call Type	GET
Parameter	**filepath** The full path to the media file whose partial offsets are being requested. **tcin** Timecode of the first frame requested **tcout** Timecode of the last frame requested **fileframerate** Frame rate, as returned in the file status report **part_file** Full UNC path to partial restored file fragment **out_filename** Output file name for partial media file (care should be taken that this does not clash with other part restores, e.g. from other sections of the same source file) *This should not have an extension, as this will added automatically*
Returns	See section below – Returned XML

EXAMPLE

```
http://PFRIndexer:60792/partialfile?filepath=\\host\share\file.mxf&tcin=00:01:00:00&tcout=00:02:00:00&fileframerate=25&part_file=\\host\media\temp\partfile&out_filename=MyRestore
```

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.

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 out_filename value passed in the Partial File Request *This is Case Sensitive*
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”/>
```