

Binary Data on External File Share

12/22/2023 1:28 pm EST

Contents

Background Information

- Writes
- Reads
- AppServer
- SignalHandler
- Operator Workstation
- Media Gateway
- Telservers
- Considerations

Setup and Troubleshooting

- MOption
- Troubleshooting

Background Information

Writes

Writes header info to SQL in the column where the binary data would have been written

Writes header info to file

Writes data to end of file

[Return to Contents](#)

Reads

Reads header info from SQL from the column where the binary data would have been written

Reads header info from file and compares to header from SQL

Reads rest of data from file

[Return to Contents](#)

AppServer

Uses Reads and Writes as described above

[Return to Contents](#)

Signal Handler

Uses Reads and Writes as described above

If the first clip then we just write it out

If the middle or last clip then we read the existing wav, append the audio in memory, and write it out (overwriting the old one)

[Return to Contents](#)

Operator Workstation

Read for wav (AppServer call) every X seconds (X = whatever the update timer is set to in the UI. Defaults to 5 seconds for wav files)

Write wav file to local hard drive

Set the slider and position of where we last were and start playing if not paused

[Return to Contents](#)

Media Gateway

Talks to Telservers - Monitor extension as "answer and record" - sends slice time and max record time

Gets messages from Telservers about recordings and where they are on the hard drive.

Send first clip as a "first" wav audio type to XML FEP driver.

Send middle clips as "middle" wav audio type to XML FEP driver.

Send last clip as "last" wav audio type to XML FEP driver.

[Return to Contents](#)

Telservers

Starts SoloProtect session.

Gets slice time and max record time.

Gets "answer and record" command.

Phone rings, we answer and start recording.

When we reach the slice time or max record time we stop recording, write the file to the hard drive and send a message to MG that we have a file and where it is located. If not max record time then start recording again.

[Return to Contents](#)

Considerations

- Once the system is set it up this way, they can't easily go back.
- We don't currently have an easy way to migrate existing binary over. We would have to consider that custom work and quote it if they want to do that.
- This will store ALL BLOG data in a separate location. It does not discriminate what that data is. We can't just do Video.
- They will have to manage the storage, backup, and maintenance of the other storage location.
- They will have to insure the bandwidth and latency of the connection to the other storage location. Delay here could cause performance issues.

[Return to Contents](#)

Setup and Troubleshooting

MOption

This functionality is controlled by several MOPTIONs. These must be updated directly in the database. There is intentionally no UI to do this because it should not be used lightly.

MOPTION 86 OPTCHAR value controls where the external files will be stored.

MOPTION 13002 OPTINT value controls the number of retries when trying to read the external files.

MOPTION 13003 OPTINT value controls the number of milliseconds between retries when retrying to read the external files.

[Return to Contents](#)

Troubleshooting

If reads/writes from/to the external file have issues, these are some messages you might see in the logger:

WriteSqlBinaryData() failed

File buffer not set.

Failed (%ld) to open file '%s' for writing.

Failed (%ld) to write binary header to file '%s'.

Failed (%ld) to set binary file pointer in '%s'.

Failed (%ld) to write binary data to file '%s'.

Failed (%ld) to delete file '%s'.

Failed (%ld) to create directory '%s'.

Failed to generate binary file name.

Failed to generate binary file path.

Failed (%ld) to open file '%s' for reading.

Failed (%ld) to read binary file header from '%s'.

Invalid binary file header in '%s'.

Binary header had an invalid checksum in '%s'

Binary header checksum in '%s' did not match the database.

Binary file id in '%s' did not match the database.

Failed to compare binary file id for '%s'.

Failed (%ld) to set binary file pointer in '%s'.

Binary file buffer is too small.

Failed (%ld) to read binary data from '%s'.

Base path to binary files too long.

Failed to generate binary directory path.

Binary file path is too long.

Binary header had an invalid checksum.

Invalid binary file header.

Failed to generate binary file unique id.

Failed to generate binary directory path.

Binary file path is too long.

Binary file base directory is not set.

Binary folder name is too long.

[Return to Contents](#)