MVX2File

Generated by Doxygen 1.8.16

Chapter 1

Mantis Vision: MVX2File

A plugin providing services/filters for writing and reading MVX data to/from MVX-formatted binary files.

Table of Contents

Chapter 2

Release Notes

1.0.0

Initial version.

Plugin

- **1.0.0_P1** | introduced `TargetMVX2FilePerStreamSnapshotWriter` filter

Documentation

- **1.0.0_D1** | added 'release notes' section

Build support

- **1.0.0_BS1** | introduced config files specifying collection of all files (libraries, assets, ...) associated with the plugin - one SuperNetwork.cfg file per each combination of OS, architecture and build configuration, located next to the plugin file (dll/so/...)

2.0.0

Plugin

- **2.0.0_P1** | updated for `Mvx2` framework version 3.0.0
- **2.0.0_P2** | fixed source filters to report correct number of 'complete' frames in files (i.e. excluding incomplete frames)
- **2.0.0_P3** | introduced alternative writing filters, which perform writing operation from standalone writing thread asynchronously:
  - `TargetMVX2FilePerStreamSnapshotWriterAsync` (as alternative to `TargetMVX2FilePerStreamSnapshotWriter`),
  - `TargetMVX2FilePerStreamWriterAsync` (as alternative to `TargetMVX2FilePerStreamWriter`),
  - `TargetMVX2FileSnapshotWriterAsync` (as alternative to `TargetMVX2FileSnapshotWriter`),
  - `TargetMVX2FileWriterAsync` (as alternative to `TargetMVX2FileWriter`)

Build support

- **2.0.0_BS1** | size of Android and LuminOS libraries reduced by ∼90%
- **2.0.0_BS2** | android API level raised from 19 to 21
- **2.0.0_BS3** | Linux and MacOS binaries do not consist of a versioned plugin file and a version-neutral symlink file anymore - the plugin file itself has version-neutral name

3.0.0

Plugin

- **3.0.0_P1** | updated for `Mvx2` framework version 4.0.0

3.1.0

Plugin

- **3.1.0_P1** | updated `MVCommon` 3rdparty dependency to version 3.0.0
- **3.1.0_P2** | updated for `Mvx2` framework version 5.0.0
- **3.1.0_P3** | fixed a bug related to **"MVX File Path"** parameter of all writer filters, which was causing multiple misbehaviours:
  - a file at the path could be missing an atom,
  - in particular use cases, when the path was changed by a user, a file at the previous path could be rewritten by a single-atom stream (discarding thus recorded data)
- **3.1.0_P4** | logs generated by the plugin use custom `MVX2File` tag instead of the generic (`MVX2`) one

3.2.0

Plugin

- **3.2.0_P1** | introduced **"Buffered frames count"** parameter to TargetMVX2FilePerStreamSnapshotWriterAsync
- **3.2.0_P2** | introduced **"Buffered frames count"** parameter to TargetMVX2FilePerStreamWriterAsync
- **3.2.0_P3** | introduced **"Buffered frames count"** parameter to TargetMVX2FileSnapshotWriterAsync
- **3.2.0_P4** | introduced **"Buffered frames count"** parameter to TargetMVX2FileWriterAsync

3.3.0

Plugin

- **3.3.0_P1** | updated `MVCommon` 3rdparty dependency to version 4.0.0
- **3.3.0_P2** | updated for `Mvx2` framework version 6.0.0
- **3.3.0_P3** | upgraded multiple internal dependencies with possible effect on all reading and writing filters

Documentation

- **3.3.0_D1** | introduced PDF documentation as an alternative to the HTML one:
  - `doc/MVX2File.pdf`

Chapter 3

Source filters

3.1 SourceMVX2FileReader

A source filter for reading frames from an MVX-formatted binary file.

The source supports filtering of the streams stored in the file by specifying IDs of the streams to be included.

The responsibility of the source is also loosening/duplication of the data optimized by writers of the plugin from the first atom of each stream to all its atoms.

Type GUID

B74863F3-13CF-47A0-9F90-07718FC52AB6

Parameters

- **"MVX File Path"**
  Specifies a path of the file.
- **"Included streams"**
  Specifies a set of stream IDs to be read from the file as a comma-separated list of numbers.

3.2 SourceMVX2MultiFileAppend

A source filter for reading frames from a sequence of MVX-formatted binary files. The source reads frames from the files and pushes them to a pipeline for processing as if it would be a single file. The files represent a partitioning of data based on frame numbers. The order of frames processed depends on the alphabetical order of the files' names.

The input to the source is a directory which is supposed to contain MVX files. Names of the files are insignificant (except for their alphabetical order and that they must have the '.mvx' extension), but in order for the source to be able to process them, each file must be a valid MVX file, it must have the same data profile and hold the same collection of streams.

As frames of the files are pushed to a pipeline, their frame numbers are modified to make the illusion of a single file being processed, because ordering of frames in each of the files in the sequence normally starts at '0'.

The responsibility of the source is also loosening/duplication of the data optimized by writers of the plugin from the first atom of each stream to all its atoms.

Type GUID

757504F3-A320-43A6-BFD1-9ACE60EDEB7E

Parameters

- **"MVX files dir"**
  Specifies a path of the directory with files. The source considers all files with '.mvx' extension in this directory, in their alphabetical order.

3.3 SourceMVX2MultiFileReader

A source filter for reading frames from a set of MVX-formatted binary files.

The input to the source is a directory which is supposed to contain MVX files. Names of the files are insignificant (except that they must have the '.mvx' extension), but in order for the source to be able to process them, each file must be a valid MVX file, it must have the same data profile and each stream can only be held by a single file at most.

The responsibility of the source is also loosening/duplication of the data optimized by writers of the plugin from the first atom of each stream to all its atoms.

Type GUID

6D82A817-BAC2-4277-B599-7848F2170ADD

Parameters

- **"MVX files dir"**
  Specifies a path of the directory with files. The source considers all files with '.mvx' extension in this directory.

Chapter 4

Target filters

4.1 TargetMVX2FilePerStreamSnapshotWriter

A target filter for splitting processed frames into streams and writing snapshots of the processed frames to a standalone MVX-formatted binary file (a file per stream).

Frames stored in the files are renumbered into a sequential order and optimized, so common data of a sequence of atoms in a stream are only stored in the first atom of the stream (readers of the plugin are performing the opposite operation - they duplicate the optimized data for each atom of a stream).

Type GUID

A74FA620-53CD-461A-A512-3217F1271DC3

Parameters

- **"MVX File Path"**
  Specifies a path for the files. Paths of actual files per stream are constructed by appending '_{stream id}' suffix to the specified path. The path can be changed in runtime, in which case files at the original paths are closed and new ones at the new paths are opened for writing.
- **"Write XML"**
  Indicates whether a human-readable, abbreviated data shall be written in accompanying files with the same name as the MVX files, but with the '.xml' extension.
- **"Create thumbnail for first frame"**
  Indicates whether a thumbnail data shall be rendered and stored for the first atom of each stream.
- **"Take snapshot"**
  Fires a trigger for taking a snapshot of the next-to-be-processed frames. A certain number of frames processed after firing the trigger will be written to the files.
- **"Frames count per snapshot"**
  Defines a number of frames to write to the files when the trigger for taking a snapshot is next fired.

4.2 TargetMVX2FilePerStreamSnapshotWriterAsync

A target filter for splitting processed frames into streams and asynchronous writing snapshots of the processed frames to a standalone MVX-formatted binary file (a file per stream). Asynchronous writing means that frames are pushed to a buffer from the pipeline thread and are pulled from the buffer and written to a file from a standalone writing thread.

Frames stored in the files are renumbered into a sequential order and optimized, so common data of a sequence of atoms in a stream are only stored in the first atom of the stream (readers of the plugin are performing the opposite operation - they duplicate the optimized data for each atom of a stream).

Type GUID

BEF1EC5E-D302-4BAD-B771-88D0E404035C

Parameters

- **"MVX File Path"**
  Specifies a path for the files. Paths of actual files per stream are constructed by appending '_{stream id}' suffix to the specified path. The path can be changed in runtime, in which case files at the original paths are closed and new ones at the new paths are opened for writing.
- **"Write XML"**
  Indicates whether a human-readable, abbreviated data shall be written in accompanying files with the same name as the MVX files, but with the '.xml' extension.
- **"Create thumbnail for first frame"**
  Indicates whether a thumbnail data shall be rendered and stored for the first atom of each stream.
- **"Take snapshot"**
  Fires a trigger for taking a snapshot of the next-to-be-processed frames. A certain number of frames processed after firing the trigger will be written to the files.

- **"Frames count per snapshot"**
  Defines a number of frames to write to the files when the trigger for taking a snapshot is next fired.
- **"Buffer size"**
  A size of buffer of frames for asynchronous writing.
- **"Buffered frames count"**
  Shows the number of frames currently buffered for asynchronous writing.
- **"Drop frames when full"**
  Indicates behaviour when the buffer of frames is full. When `true`, new frames from pipeline are dropped and pipeline continues its execution. When `false`, pipeline thread is blocked until there is some free space in the buffer.
- **"Dropped frames count"**
  Shows the number of frames that were dropped because of the buffer of frames being full.
- **"Reset dropped frames counter"**
  Resets the counter of dropped frames.

4.3 TargetMVX2FilePerStreamWriter

A target filter for splitting processed frames into streams and writing each of the streams to a standalone MVX-formatted binary file (a file per stream).

Frames stored in the files are renumbered into a sequential order and optimized, so common data of a sequence of atoms in a stream are only stored in the first atom of the stream (readers of the plugin are performing the opposite operation - they duplicate the optimized data for each atom of a stream).

Type GUID

C6AB7E7D-E3E2-4E3A-ACAF-9A4DD28B318B

Parameters

- **"MVX File Path"**
  Specifies a path for the files. Paths of actual files per stream are constructed by appending '_{stream id}' suffix to the specified path. The path can be changed in runtime, in which case files at the original paths are closed and new ones at the new paths are opened for writing.
- **"Write XML"**
  Indicates whether a human-readable, abbreviated data shall be written in accompanying files with the same name as the MVX files, but with the '.xml' extension.
- **"Create thumbnail for first frame"**
  Indicates whether a thumbnail data shall be rendered and stored for the first atom of each stream.
- **"Enable Recording"**
  Enables/disables recording. When recording is re-enabled, already existing files are overwritten by new ones.

4.4 TargetMVX2FilePerStreamWriterAsync

A target filter for splitting processed frames into streams and asynchronous writing each of the streams to a standalone MVX-formatted binary file (a file per stream). Asynchronous writing means that frames are pushed to a buffer from the pipeline thread and are pulled from the buffer and written to a file from a standalone writing thread.

Frames stored in the files are renumbered into a sequential order and optimized, so common data of a sequence of atoms in a stream are only stored in the first atom of the stream (readers of the plugin are performing the opposite operation - they duplicate the optimized data for each atom of a stream).

Type GUID

9F77FB3F-F205-491D-A12C-14B3266F700B

Parameters

- **"MVX File Path"**
  Specifies a path for the files. Paths of actual files per stream are constructed by appending '_{stream id}' suffix to the specified path. The path can be changed in runtime, in which case files at the original paths are closed and new ones at the new paths are opened for writing.
- **"Write XML"**
  Indicates whether a human-readable, abbreviated data shall be written in accompanying files with the same name as the MVX files, but with the '.xml' extension.
- **"Create thumbnail for first frame"**
  Indicates whether a thumbnail data shall be rendered and stored for the first atom of each stream.
- **"Enable Recording"**
  Enables/disables recording. When recording is re-enabled, already existing files are overwritten by new ones.
- **"Buffer size"**
  A size of buffer of frames for asynchronous writing.
- **"Buffered frames count"**
  Shows the number of frames currently buffered for asynchronous writing.
- **"Drop frames when full"**
  Indicates behaviour when the buffer of frames is full. When `true`, new frames from pipeline are dropped and pipeline continues its execution. When `false`, pipeline thread is blocked until there is some free space in the buffer.
- **"Dropped frames count"**
  Shows the number of frames that were dropped because of the buffer of frames being full.
- **"Reset dropped frames counter"**
  Resets the counter of dropped frames.

4.5 TargetMVX2FileSnapshotWriter

A target filter for writing snapshots of processed frames to an MVX-formatted binary file.

Frames stored in the file are renumbered into a sequential order and optimized, so common data of a sequence of atoms in a stream are only stored in the first atom of the stream (readers of the plugin are performing the opposite operation - they duplicate the optimized data for each atom of a stream).

Type GUID

5F18B69C-472C-4BF7-902C-593A24A53267

Parameters

- **"MVX File Path"**
  Specifies a path for the file. The path can be changed in runtime, in which case a file at the original path is closed and a new one at the new path is opened for writing.
- **"Write XML"**
  Indicates whether a human-readable, abbreviated data shall be written in an accompanying file with the same name as the MVX file, but with the '.xml' extension.
- **"Create thumbnail for first frame"**
  Indicates whether a thumbnail data shall be rendered and stored for the first atom of each stream.
- **"Take snapshot"**
  Fires a trigger for taking a snapshot of the next-to-be-processed frames. A certain number of frames processed after firing the trigger will be written to the file.
- **"Frames count per snapshot"**
  Defines a number of frames to write to the file when the trigger for taking a snapshot is next fired.

4.6 TargetMVX2FileSnapshotWriterAsync

A target filter for asynchronous writing snapshots of processed frames to an MVX-formatted binary file. Asynchronous writing means that frames are pushed to a buffer from the pipeline thread and are pulled from the buffer and written to a file from a standalone writing thread.

Frames stored in the file are renumbered into a sequential order and optimized, so common data of a sequence of atoms in a stream are only stored in the first atom of the stream (readers of the plugin are performing the opposite operation - they duplicate the optimized data for each atom of a stream).

Type GUID

990F8140-8912-464F-B85A-6AB300AD0E37

Parameters

- **"MVX File Path"**
  Specifies a path for the file. The path can be changed in runtime, in which case a file at the original path is closed and a new one at the new path is opened for writing.
- **"Write XML"**
  Indicates whether a human-readable, abbreviated data shall be written in an accompanying file with the same name as the MVX file, but with the '.xml' extension.
- **"Create thumbnail for first frame"**
  Indicates whether a thumbnail data shall be rendered and stored for the first atom of each stream.

- **"Take snapshot"**
  Fires a trigger for taking a snapshot of the next-to-be-processed frames. A certain number of frames processed after firing the trigger will be written to the file.
- **"Frames count per snapshot"**
  Defines a number of frames to write to the file when the trigger for taking a snapshot is next fired.
- **"Buffer size"**
  A size of buffer of frames for asynchronous writing.
- **"Buffered frames count"**
  Shows the number of frames currently buffered for asynchronous writing.
- **"Drop frames when full"**
  Indicates behaviour when the buffer of frames is full. When `true`, new frames from pipeline are dropped and pipeline continues its execution. When `false`, pipeline thread is blocked until there is some free space in the buffer.
- **"Dropped frames count"**
  Shows the number of frames that were dropped because of the buffer of frames being full.
- **"Reset dropped frames counter"**
  Resets the counter of dropped frames.

4.7 TargetMVX2FileWriter

A target filter for writing processed frames to an MVX-formatted binary file.

Frames stored in the file are renumbered into a sequential order and optimized, so common data of a sequence of atoms in a stream are only stored in the first atom of the stream (readers of the plugin are performing the opposite operation - they duplicate the optimized data for each atom of a stream).

Type GUID

972EE495-EB8E-48A8-AB94-43797E34A436

Parameters

- **"MVX File Path"**
  Specifies a path for the file. The path can be changed in runtime, in which case a file at the original path is closed and a new one at the new path is opened for writing.
- **"Write XML"**
  Indicates whether a human-readable, abbreviated data shall be written in an accompanying file with the same name as the MVX file, but with the '.xml' extension.
- **"Create thumbnail for first frame"**
  Indicates whether a thumbnail data shall be rendered and stored for the first atom of each stream.
- **"Enable Recording"**
  Enables/disables recording. When recording is re-enabled, an already existing file is overwritten by a new one.

4.8 TargetMVX2FileWriterAsync

A target filter for asynchronous writing processed frames to an MVX-formatted binary file. Asynchronous writing means that frames are pushed to a buffer from the pipeline thread and are pulled from the buffer and written to a file from a standalone writing thread.

Frames stored in the file are renumbered into a sequential order and optimized, so common data of a sequence of atoms in a stream are only stored in the first atom of the stream (readers of the plugin are performing the opposite operation - they duplicate the optimized data for each atom of a stream).

Type GUID

3802D826-7D62-41BF-A92E-6A56E3700590

Parameters

- **"MVX File Path"**
  Specifies a path for the file. The path can be changed in runtime, in which case a file at the original path is closed and a new one at the new path is opened for writing.
- **"Write XML"**
  Indicates whether a human-readable, abbreviated data shall be written in an accompanying file with the same name as the MVX file, but with the '.xml' extension.
- **"Create thumbnail for first frame"**
  Indicates whether a thumbnail data shall be rendered and stored for the first atom of each stream.
- **"Enable Recording"**
  Enables/disables recording. When recording is re-enabled, an already existing file is overwritten by a new one.
- **"Buffer size"**
  A size of buffer of frames for asynchronous writing.
- **"Buffered frames count"**
  Shows the number of frames currently buffered for asynchronous writing
- **"Drop frames when full"**
  Indicates behaviour when the buffer of frames is full. When `true`, new frames from pipeline are dropped and pipeline continues its execution. When `false`, pipeline thread is blocked until there is some free space in the buffer.
- **"Dropped frames count"**
  Shows the number of frames that were dropped because of the buffer of frames being full.
- **"Reset dropped frames counter"**
  Resets the counter of dropped frames.