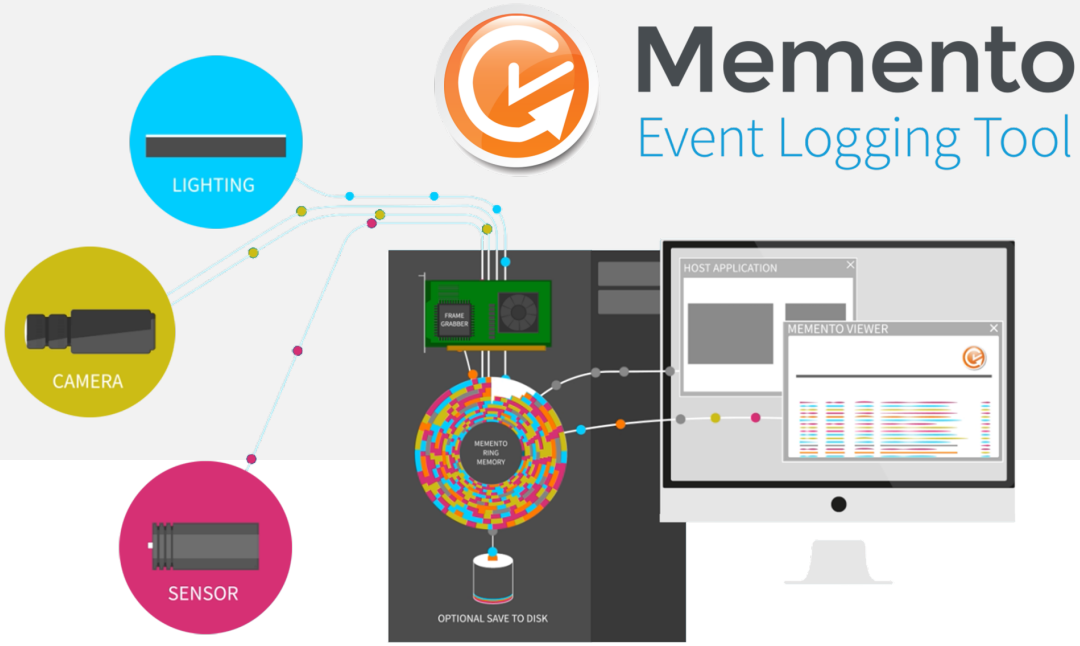
EURESYS™
Excellence in vision

GETTING STARTED

Memento

Memento 12.0.0



Terms of Use

EURESYS s.a. shall retain all property rights, title and interest of the documentation of the hardware and the software, and of the trademarks of EURESYS s.a.

All the names of companies and products mentioned in the documentation may be the trademarks of their respective owners.

The licensing, use, leasing, loaning, translation, reproduction, copying or modification of the hardware or the software, brands or documentation of EURESYS s.a. contained in this book, is not allowed without prior notice.

EURESYS s.a. may modify the product specification or change the information given in this documentation at any time, at its discretion, and without prior notice.

EURESYS s.a. shall not be liable for any loss of or damage to revenues, profits, goodwill, data, information systems or other special, incidental, indirect, consequential or punitive damages of any kind arising in connection with the use of the hardware or the software of EURESYS s.a. or resulting of omissions or errors in this documentation.

This documentation is provided with Memento 12.0.0 (doc build 6019).

Contents

Documentation Updates

The documentation updates for Memento 12.0.0 are detailed in the table below.

Update description	Section
New kinds: Studio, Sockets and GigE Vision (for future use).	See section "Kind attribute" on page 8

1. Product Presentation

Memento
Event Logging Tool

Memento is an advanced system for logging event messages. It greatly facilitates the debugging of machine vision applications using Euresys frame grabbers. It is non-intrusive as the required CPU load is extremely low.

The Memento system is made up of three main components: the Memento driver, the Memento application, and the Memento contributor(s).

Memento is a set of software tools allowing:

- frame grabber drivers and user space applications – Memento Contributors – to inject trace messages - Memento Traces – into a common memory area – the Memento Ring Buffer.
- Memento contributors to time-stamp the Memento messages using a common time scale – Memento Time Scale.
- to view selected sets of recent or past Memento traces in the Memento application (GUI or Console mode).
- to analyze the occurrence of events and state changes in the digital image acquisition processes, which allows to measure latencies and detect defects during acquisition.
- to dump Memento data from the Memento ring buffer to disk using the Memento Dump function.

The Memento software package has two main components:

- a kernel-mode driver: Memento driver
- a user-mode application: Memento application

2. Concepts

2.1. Contributors

The Memento contributors are Euresys kernel drivers and Euresys or third-party user space applications embedding Memento traces at particular code locations.

A Memento trace is a non-blocking and lightweight software macro that, upon execution, builds up a message and pushes it into the ring buffer.

Coaxlink Driver & Libraries

The Coaxlink driver software package is a major message contributor. For instance, Coaxlink Driver 4.7 includes ~1,000 distinct instances of Memento traces!

All the Coaxlink products implement **Memento-in-hardware**. This feature allows hardware events to be logged in Memento with an accurate time attribute.

MultiCam Driver & Libraries

The kernel-mode drivers for Grablink Base, Grablink DualBase, Grablink Full, and Grablink Full XR products as well as the MultiCam and clseremc libraries can also contribute to Memento.

User Application

A user application may include Memento traces capable of generating messages having a custom content and two attributes: `kind` and `level`.

Any severity level can be assigned to the `level` attribute.

Sixteen (16) distinct values are available for the `kind` attribute:`user0`, … `user9`, … `userA`, … `userF`.

NOTE
The value `user` is kept for backward compatibility: it is equivalent to `user0`.

EGrabber

The EGrabber API layer generates Memento Traces with the `Egrabber` kind attribute.

2.2. Messages

A Memento message is composed of a **message body** and several **message attributes**.

Message body

The message body is a text string composed of a **fixed frame** and optional **arguments**. The arguments are used as placeholders for variable context information.

Message body examples:

- The following message body has no arguments: `GenICam module has been unloaded`
- The following message body has two arguments, the date and the time: `Time: 2016-04-22 14:43:29.0493338 UTC`

Message attributes

Message attributes are used for qualifying the message body:

- Mandatory attributes
  - The **Kind** and **Level** attributes are mandatory.
  - They must always be defined by the Memento contributors.
- Optional attributes
  - The **Time**, **Process ID**, **Thread ID**, **Card ID**, **Connector ID** and **Stream ID** attributes are optional.
  - The Memento contributors decide whether or not to use them.

  In practice, these optional attributes are always used when their values can be retrieved and when this information makes sense in the context.

Level attribute

The `Level` attribute defines the **severity level** of the message on a scale having 7 steps:

Value	Severity	Description
Critical	Highest	Unrecoverable error message
Error	Higher	Error message
Warning	High	Warning message
Notice	Medium	Notification message
Info	Low	Information message
Debug	Lower	Debug message
Verbose	Lowest	Nattering

> **NOTE**
> **Debug** and **Verbose** levels are used for messages intended for Euresys engineering and technical support.

Kind attribute

The mandatory **Kind** attribute defines the kind, nature, origin … of the message:

Value	Description
API	API function calls
Acquisition	Messages related to image acquisition (e.g., data stream start/stop, start/end of scan)
CIC	Messages related to the Camera and Illumination Controller
CoaXPress	Messages related to CoaXPress protocol (e.g., device discovery, control messages)
CustomLogic	Messages emitted by the CustomLogic core
DMA	Messages related to DMA (Direct Memory Access) transfers
DPC	Messages related to DPC (Deferred Procedure Calls) queued by the driver when it receives an interrupt request
Descriptors	Messages related to DMA descriptors
EEPROM	Messages related to on-board EEPROM (where part number, serial number and OEM key are stored)
Egrabber	Messages produced by EGrabber
Event	Messages related to the event notification infrastructure
FFC	Messsages related to Flat Field Correction
Firmware	Messages related to the firmware variant
Flash	Messages related to the on-board flash memory (where the firmware image is stored)
FPGA	Messages related to: • Firmware image • CIC (Camera and Illumination Controller)
GenAPI	Messages related to Euresys GenAPI implementation
GigE Vision	Messages related to the GigE Vision protocol
Hardware	Messages emitted by the frame grabber firmware
I2C	Messages related to the I2C communication between 2 frame grabber components

Value	Description
IOCTL	Messages related to the communication between user-mode software components and the kernel driver
IRQ	Messages related to the interrupt requests received by the kernel driver
Licensing	Messages related to the licensing infrastructure
LUT	Messages related to the look-up table processing
MIO	Messages related to the I/O extension module
Memento	Messages related to Memento
Onboard Memory	Messages related to the onboard memory
PCI Express	Messages related to the PCI Express configuration space (e.g., vendor and device IDs, capabilities)
PnP	Messages related to the interaction between the driver and the Windows Plug and Play Manager
Power	Messages related to the interaction between the driver and the Windows Power Manager
Profiling	A small number of messages for doing profiling (IRQ counter, time spent with interrupts masked, DPC counter, time spent in DPC, buffer push/pop and DMA transfer completion)
SPI	Messages related to the SPI interface (used to access the on-board flash)
Script	Messages related to the GenApi scripts
Serial Line	Messages related to the Camera Link serial communication
Socket	Messages related to the Sockets interface
Studio	Messages related to the Euresys Studio application
Thread	Messages related to worker threads used by the driver
Time	Messages related to system data and time
Timer	Messages related to operating system timers
User0 … User9, UserA … UserF	Messages emitted by the user application

The above list is open; new values can be appended in future Memento releases.

Time attribute

For all but the Hardware kind of messages, the **time** attribute defines the time of occurrence on the Memento time scale of the creation of the message by the contributor.

For Hardware kind of messages, the **time** attribute defines the time of occurrence on the Memento time scale of the hardware event at the origin of the message.

These messages are prefixed by [ts:HARDWARE_TIMESTAMP] where HARDWARE_TIMESTAMP is the time of occurrence on the *card local time scale*.

NOTE
The time of occurrence reported by the hardware using the card local time is converted to Memento time.

Process ID & Thread ID attributes

The **Process ID** attribute reports the process identifier of the process producing the message.

The **Thread ID** attribute reports the thread identifier of the thread producing the message.

Card ID, Connector ID and Stream ID attributes

The **Card ID** attribute reports the identifier of the frame grabber card the message belongs to. This is a zero-based index assigned by the corresponding card driver.

- The **Card ID** is global within a specific driver.
- Coaxlink cards have a unique driver for all types. If, for instance, a system has 1 Coaxlink Mono and 1 Coaxlink Duo, there will be Memento traces related to Coaxlink Card ID #0 and #1.
- Grablink cards have a specific driver for each card type. If, for instance, a system has 2 Grablink Base and 2 Grablink DualBase , there will be Memento traces related to Grablink Base Card ID #0 and #1 as well as Grablink DualBase card ID #0 and #1.

The **Connector ID** attribute reports the identifier of the camera connector the message belongs to.

- For Coaxlink cards, the **Connector ID** is the ID (from A to H) of the CoaXPress host interface connector.
- For Grablink cards, the **Connector ID** is the ID (M, A, B) of the Camera Link connector.

The **Stream ID** attribute reports the identifier of the image data stream the message is related to.

- For Coaxlink cards, the **Stream ID** is the ID (0, 1, 2, 3) of the image data stream transferred from the camera to the frame grabber.
- For Grablink cards, the **Stream ID** is the ID (0) of the image data stream transferred from the camera to the frame grabber.

2.3. Time Scale

The Memento time scale is a common time scale used for time-stamping all the Memento messages produced by the Memento contributors within a Host PC.

Having a common time scale is a key feature of Memento since it allows messages from different origins to be time-coordinated.

The Memento time scale:

- resets at the cold boot of the Host PC.
- increments by 1 every microsecond.
- never overflows thanks to a 64-bit quantification.

TIP
Memento uses the *Performance Counter* of the Host PC to build the Memento time scale. Consequently, its accuracy and its stability are Host PC dependent.

NOTE
For Windows users only: The Memento time scale is not reset when Windows performs a Fast Boot, even after a Power Off. When the Fast Boot option is activated, the only way to reset the Memento time scale is to perform a Windows Restart.

UTC vs. Memento time

When a Coaxlink card is installed in the Host PC, the Coaxlink Driver generates every minute a UTC time message: `Time: 2016-04-22 14:43:29.0493338 UTC`

Such message allows for establishing the relationship between the UTC (Coordinated Universal Time) and the Memento time.

The time reported in the above message is actually the Host PC time of the local time zone converted to the UTC time zone.

The accuracy of the reported UTC time depends on the Host PC. When the Host PC implements a synchronization mechanism to Internet time servers, the UTC time accuracy is pretty good.

2.4. Memento Driver

The kernel-mode Memento driver is responsible for managing the **Memento Ring Buffer**: a region of the host PC memory where the **Memento Contributors** store the **Memento Messages**.

The Coaxlink driver and/or the MultiCam Driver automatically connects to the Memento driver.

Once the connection is established, the messages composed by the contributors are first screened by the **Ring Buffer Filter** according to the **Ring Buffer Configuration** settings.

Only the messages that successfully pass through the ring buffer filter are effectively written into the ring buffer.

Ring Buffer

The Memento ring buffer is an area of the Host PC memory where the Memento messages are written by the contributors.

The Memento ring buffer:

- has a fixed capacity of 524.288 (512 K) message slots of 64 bytes each.
- is managed as a "ring buffer": A new message replaces the oldest one when it becomes full.

Message compacting

To optimize the storage and the throughput efficiencies of the Memento ring buffer, Memento messages issued by Euresys drivers or libraries are stored in a compact form.

A compacted message without arguments occupies only a single 64-byte memory slot in the ring buffer. When the message has arguments, it requires additional memory slots.

The original messages are reconstructed by the **Memento** application.

TIP
Messages issued by user applications are not compacted and thus, require more space in the Memento ring buffer.

Sequential number

The Memento driver assigns a sequential number to every message entering the Memento ring buffer.

The sequential number is a 64-bit positive integer number that increments by 1 on every message input. The sequential number is not reset every boot session.

The sequential number unambiguously identifies a Memento message in the whole message history of a Host PC.

Ring buffer persistence (Windows only)

The Memento driver maintains a synchronized copy of the Memento ring buffer on disk.

This allows the state and the content of the Memento ring buffer to be restored at the next boot session after a shutdown.

> **TIP**
> This mechanism operates correctly when the current session terminates normally. If the current session terminates with a "blue screen", or if the reset button is pushed, or if the power supply is removed, the Memento ring buffer data is appended to the **crash dump** file.

Ring Filter

The **Ring Filter** is applied to the message flow at the entry point of the Memento ring buffer.

The ring filter checks the `level` attribute of all the incoming messages against the **Ring Filter** rules and decides if the message has to be logged or not into the Memento ring buffer.

Ring filter rules

The ring filter rules set is composed with:

- One **default rule**
- Zero or more **kind-specific rules**

Default Rule

The default rule has one argument: `level`.

The default rule applies only to messages having a `kind` attribute that no kind-specific rule is defined for.

If no kind-specific rule is defined for the **DPC** kind, for example, the default rule will apply to the messages with **DPC** kind.

Kind-specific Rules

A kind-specific rule has two arguments: `kind` and `level`.

The rule applies only to messages having the specified `kind` attribute.

The kind-specific rules supersede the default rule.

Ring Filter Action

Are logged into the Memento ring buffer:

- any message having no matching kind-specific rule and having a severity level greater than or equal to the specified level in the default rule.
- any message having a matching kind-specific rule and having a severity level greater than or equal to the specified level in the matching kind-specific rule.

The other messages are rejected and lost forever.

Level Filtering

Rule Level	Accepted Message Levels
Critical	Critical
Error	Critical Error
Warning	Critical Error Warning
Notice	Critical Error Warning Notice
Info	Critical Error Warning Notice Info
Debug	Critical Error Warning Notice Info Debug
Verbose	Critical Error Warning Notice Info Debug Verbose

Ring Buffer Configuration

The Memento configuration consists in filtering the traces to be added to the ring buffer. It relies on the ring filter rules. It is therefore also called the ring buffer configuration.

Memento configuration change

The Memento configuration settings can be changed at any time and take effect immediately.

Memento configuration persistence (Windows only)

The Memento configuration settings are persistent. The settings are stored into a configuration file on disk and are automatically restored at the next boot session.

Default Memento configuration

The default Memento configuration defines the following rule filters:

- For **Profiling** kind, all messages with verbosity level **Notice** and higher are injected by default.
- For **DPC**, **IOCTL**, **IRQ** and **Memento** kinds, all messages with verbosity level **Info** and higher are injected by default.
- For **CoaXPress**, **GenApi** and **PnP** kinds, all messages with verbosity level **Debug** and higher are injected by default.
- For all other kinds, the default rule applies: all messages with verbosity level **Verbose** and higher are injected.

2.5. Application Interfaces

Memento is a multi-mode application. Possible modes are:

Short Name	Full Name & Description
`gui`	**Graphical User Interface** (Windows default mode) A single-window graphical user interface application to: • Show/Change the Memento configuration • Monitor graphically the input activity of the Memento ring buffer • View a selected set of messages of the ring buffer • Control interactively the selection of messages to display • Dump the ring buffer data to disk Windows default mode.
`config`	**Console Configurator** A command prompt console application to: • Show/Change the Memento configuration
`view`	**Console Viewer** A command prompt console application to: • View a selected set of messages of the ring buffer Linux default mode.
`dump`	**Console Dumper** A command prompt console application to: • Dump the ring buffer data to disk

Multiple instances of the Memento application can run concurrently.

2.6. Memento Application

You can retrieve selectively the Memento traces from the Memento ring buffer and view in the Memento application.

The Memento application is available in two modes:

- Graphical user interface
- Console viewer

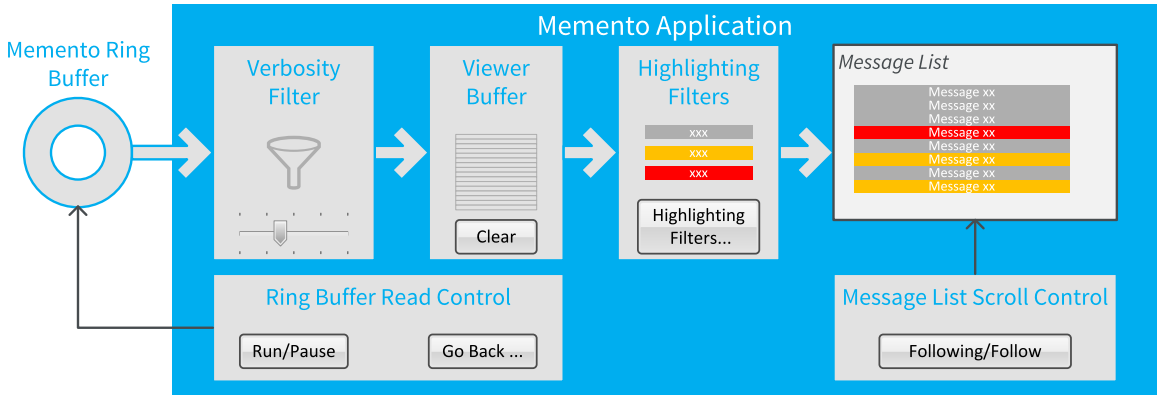
The Memento application helps you among others to:

- filter traces to display and highlight the most relevant events in your processes.
- search for traces based on their content or their time stamp.
- rapidly spot errors or critical traces on a plot showing the recent activity and on a plot displaying the number of events according to their criticality.

- investigate an issue by analyzing process-related events and state changes displayed as waveforms.
- save traces from the current process to a dump file for further investigation.

Block Diagram

The following drawing shows the function blocks of the Memento application.



Memento Application – VIEW Mode Function Blocks and Data Flow

The Memento application processes the messages as follows:

- It reads messages from the Memento ring buffer.
- It stores the messages satisfying the rules of the verbosity filter into the viewer buffer.
- It reads messages from the viewer buffer into the Memento application.
- It highlights (or possibly hides) the messages according to the settings of the highlighting filters.
- It displays the highlighted messages in the message list of the Memento application (GUI or console).

Viewer Buffer

The viewer buffer is a memory area where the Memento application stores the messages to display.

The viewer buffer is cleared when starting up the Memento application. Then it accumulates all the incoming messages from the ring buffer that satisfy the rules of the verbosity filter.

The viewer buffer has a capacity of 250,000 message slots. This can be modified with the command line argument "history", a higher value means a higher memory consumption.

The viewer buffer can be cleared at any time by the user using the **Clear** control in the Memento application.

Buffer controls

By default, the viewer buffer is filled with the most recent messages of the Memento ring buffer.

However, other options are available using the **Run|Pause** and the **Go Back** controls.

Run/Pause controls

The **Run/Pause** control allows you to pause the extraction of the messages from the Memento ring buffer and the filling process of the viewer buffer.

After a pause, the message extraction resumes.

NOTE
In the meantime messages can be lost if the maximum ring capacity has been reached.

Go Back control

The **Go Back** control allows to reload the viewer buffer with older messages of the Memento ring buffer.

Verbosity Filter

The **Verbosity Filter** is applied in the message flow between the Memento ring buffer and the viewer buffer.

The verbosity filter allows you to display only the messages having a verbosity level (`level` attribute) equal to or greater than the defined verbosity level.

Verbosity Level	Accepted Message Severity Levels
Critical – *Lowest verbosity*	Critical – *Highest severity*
Error	Critical Error
Warning	Critical Error Warning
Notice	Critical Error Warning Notice
Info	Critical Error Warning Notice Info
Debug	Critical Error Warning Notice Info Debug
Verbose – *Highest verbosity*	*All levels accepted*

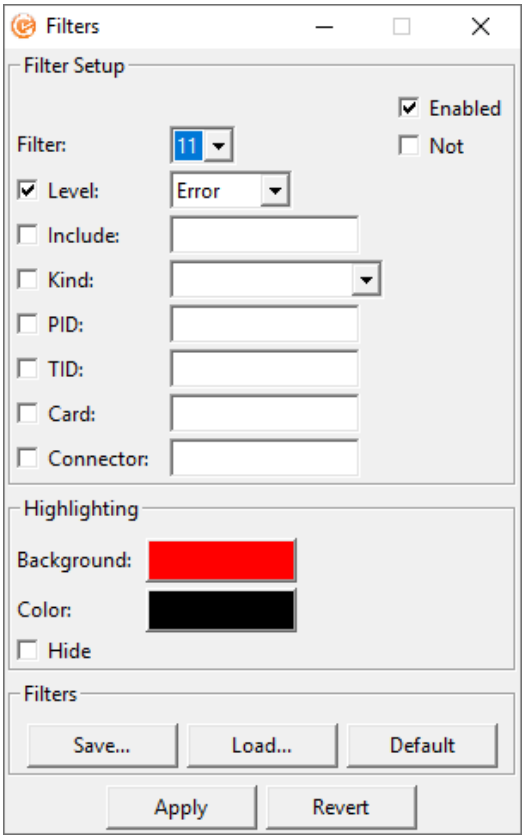
The default verbosity level is **Notice**.

Highlighting Filters

In the Memento application, sixteen **Highlighting Filters** are available to apply layout options (highlight and/or font color or hide) to the traces that meet the filter rule defined in a given filter. These filters are applied after the verbosity filter.

Each highlighting filter has two sets of controls:

- The controls for the **filter rule** that select a specific set of messages.
- The controls for the **filter action** that define the actions to be performed on the selected set of messages.



Filter Rule

The rule of an **Highlighting Filter** is defined using the following settings:

- An enable control
- A polarity control
- Seven filtering criteria (six based on attribute values and one based on the message content).

See the section Highlighting or Hiding Traces in the User Guide (D603) for more information on the filtering rule.

Filter Action

The action of an **Highlighting Filter** is defined using the following settings:

- A hide control
- A background color selector
- A font color selector

The background color and font color can be set simultaneously.

The hide action excludes the other actions.

See the section Highlighting or Hiding Traces in the User Guide (D603) for more information on possible actions.

Working principles

- Sixteen **Highlighting Filters** are available in the Memento application.
- Only enabled filters are taken into account.
- When a trace meets a filter rule, the layout options for that filter are applied. Subsequent filters are disregarded for this trace.
- The Memento application parses the filters from filter 1 to filter 16.
- By default, the filters from 11 to 16 are enabled. Each of them is configured to apply predefined layout options for a given verbosity level.
- As a consequence, a user-defined filter (1 to 10) has priority over the default filters (11-16).

Message List

The **Memento** application composes the message list with messages coming from the viewer buffer through the highlighting filters.

The messages are sorted by increasing time order if the time attribute is available otherwise by sequential order of arrival into the Memento ring.

Messages hidden by the highlighting filter are not displayed in the message list.

Scroll Control

By default the message list is updated as soon as a new message arrives in the viewer buffer: the message list follows the viewer buffer. This mode of operation is named **auto-scroll**.

This auto-scroll mode can be disabled to allow older messages of the message list to be read.

In the Memento application, the **Follow|Following** toggle button enables or disables the **auto-scroll**.

2.7. Dump

You can save the data from the Memento ring buffer into a dump file using the Memento dump function of the Memento application.

This function is available in both the GUI and console modes of the Memento application.

You can also directly start a dump file from the Windows Start menu.

Start a dump file from the Windows start menu

Without opening the Memento application, you can start saving the data added subsequently to the ring buffer into a dump file as follows:

- From the Windows Start menu, select the command **Euresys Memento > Start Memento Logging**.

When you select this option, the Memento application:

- creates a dump file `dumXXXX.memento` saved to `%USERPROFILE%\memento\` where XXXX is an automatically generated alphanumeric string.
- opens a command line window that displays the dump file size.
- feeds the dump file continuously with the subsequent data from the ring buffer as long as the command line window remains open.

Dump files from the Memento application

By default, the Memento dump function dumps the next-to-come data from the ring buffer.

However, other options are available using the **Rewind** and the **Boot Session** options.

For more information on the dump function, refer to the section Saving Traces in the user guide or Dump Options in the reference manual.

3. Setting up Memento

3.1. Installing Memento

Memento is distributed on the support section of the Euresys website:
https://www.euresys.com/Embedded-Vision-Software-Drivers-and-Documentation.

NOTE
The Euresys website download area may require user authentication. The user ID and password are defined by the user. Access is free and unrestricted.

1. Select the *Coaxlink product series* to display the file list corresponding to the latest available Coaxlink driver and Memento releases or select the *Grablink product series* to display the file list corresponding to the latest available MultiCam driver and Memento releases.
2. Select the setup file corresponding to your operating system and your processor architecture

memento-<OS>-<ARCH>-<MA.MI.RE.BU>.<EXT>

- The <OS> field designates the operating system: `linux`, `macos`, `win`, `win10`
- The <ARCH> field designates the processor architecture: `aarch64`, `x86`, `x86_64`
- The <MA.MI.RE.BU> designates respectively the major and minor version numbers, the revision and the build numbers of the driver package
- <The EXT> field designates the file type: `exe`, `tar.gz`, `pkg`

NOTE
For an installation on Windows 10, use the `memento-win10-<ARCH>-<MA.MI.RE.BU>.exe` setup file with drivers signed by Microsoft.

3. Launch the installer tool to install the driver files and software tools on your PC.

NOTE
If you have an existing Memento driver already installed, the installer tool prompts you to uninstall it before being able to continue. Otherwise, it prompts you for the selection of the destination folder. The default folder is `C:\Program Files\Euresys\Memento` for Windows versions.

For Linux users only

Memento must be installed prior to Coaxlink or MultiCam driver.

If the Coaxlink or MultiCam driver is already installed, proceed as follows:

1. Uninstall Coaxlink or MultiCam driver.
2. Install Memento.
3. Re-install Coaxlink or MultiCam driver.

3.2. Setting up the Driver

The connection between the Memento contributors and the Memento driver is automatically established.

When the PC hosting the Memento driver is running, the traces start being added to the ring buffer as soon as one of the contributors sends Memento traces to the ring buffer.

Default Memento configuration

The default Memento configuration is defined using so-called **Ring Filters** as it defines the filtering rules for adding traces on the ring buffer. It is also called the ring buffer configuration.

The ring filters rely on the **Kinds** and on the **Verbosity Level**:

- The kinds provide a classification based on the nature or origin of the messages – for example messages related to the firmware, to an API, to the communication between components, etc.
- The verbosity level provides a classification based on the message criticality. The levels are the following from the least verbose (most critical) to the most verbose (least critical): **Critical, Error, Warning, Notice, Info, Debug, Verbose**.

This is the default **Memento** configuration:

- For messages having the **Profiling** kind, all messages with verbosity level **Notice** and more critical are added by default.
- For messages having the **DPC**, **IOCTL**, **IRQ** and **Memento** kinds, all messages with verbosity level **Info** and more critical are added by default.
- For messages having the **CoaXPress**, **GenApi** and **PnP** kinds, all messages with verbosity level **Debug** and more critical are added by default.
- For all other message kinds, the default rule applies, that means that all messages with verbosity level **Verbose** and more critical are added.

Change the Memento configuration

The default configuration corresponds to the **Default** configuration profile. It is adequate for most use cases and prevents users from being overwhelmed with too much information. Besides this profile, a **Verbose** profile is available. It sets the maximum verbosity level for all

message kinds.

You can change the Memento configuration in several ways:

From Windows Start menu

From the Windows **Start Menu > Euresys Memento**, two options allow you to set the requested configuration profile:

- **Set Memento to the Highest Verbosity Level** to apply the **Verbose** configuration profile.
- **Reset Memento to the Default Verbosity Level** to reapply the **Default** configuration profile.

From the Memento GUI

From the Memento graphical user interface, you can change and refine the verbosity level per message kind using the **Ring Filters**.

See the section Filtering Traces in the Ring Buffer in the User Guide (D603).

From the Memento console

From the Memento console, you can apply the requested profile and change the verbosity level per message kind, using the **Configuration** command and appropriate attributes.

WARNING
On Linux and macOS, the configuration changes are not persistent: the Memento configuration reverts to the default settings at every boot.

On Windows, the configuration changes are persistent: if traces have already been added to the ring buffer in a previous session with a given Memento configuration, the latest Memento configuration is applied.

Examples

This command line sets Memento to the **Verbose** configuration profile:

- `memento config --profile=Verbose`

These command lines reset Memento to the **Default** configuration profile:

- `memento config --reset` or `memento config --profile=Default`

This command line sets specifically the **API** kind to the **Debug** verbosity, for example:

- `memento config --kind=API --verbosity=Debug`

3.3. Starting Memento

Start adding data to the ring buffer

When the PC hosting the Memento driver is running, the traces start being added to the ring

buffer as soon as one of the contributors sends Memento traces to the ring buffer.

As long as past activity is still available in the ring buffer, you will be able to view past traces in the Memento application (GUI or Console mode) even if it was closed when the traces were added.

Start the Memento application

To start the Memento application (graphical user interface), do one of the following actions:

- Click the Memento icon on the desktop.
- Select the application from the **Windows Start Menu > Euresys Memento > Memento**.
- Use the following command line from the Memento console: `memento gui --hide-console`.

Options from the Windows Start menu

From the **Windows Start Menu > Euresys Memento**, you have a quick access to some options to quickly perform a ring buffer configuration or save traces to a dump file:

Command	Description
Record Memento and XPerf traces	This generates a .zip file including the memento dump with all the data currently stored in the ring buffer, as well as the data collected by the Windows Performance Monitoring tools. This option is only relevant on Windows.
Reset Memento to Default Verbosity Level	This resets the ring buffer configuration to the **Default** profile. See Setting up the Driver in the Getting Started guide (D602).
Set Memento to Highest Verbosity Level	This sets the ring buffer configuration to the **Verbose** profile. See Setting up the Driver in the Getting Started guide (D602).
Start Memento Logging	This creates a dump file where all new data added to the ring buffer are stored. See the section Saving Traces (GUI) in the User Guide (D603) and (Console) in the Reference Manual (D604) if you want to define additional parameters for your dump file.