FeatStd V3.2.1

- FeatStd CMake Switches
- FeatStd Monitor
- FeatStd Platform
- FeatStd Util

FeatStd CMake Switches

V3.2.0	V3.2.1
FEATSTD_MONITOR_TYPE	MONITOR_COM_TYPE
FEATSTD_MONITOR_BLOCK_UNTIL_CONNECT	MONITOR_BLOCK_UNTIL_CONNECT
FEATSTD_MONITOR_TCPIP_PORT	MONITOR_TCPIP_PORT
FEATSTD_MONITOR_TCPIP_ADDRESS	MONITOR_TCPIP_ADDRESS
FEATSTD_MONITOR_TCPIP_STACK_ENABLED	MONITOR_TCPIP_STACK_ENABLED
FEATSTD_MONITOR_SERIALPORT_BAUDRATE	MONITOR_SERIALPORT_BAUDRATE
FEATSTD_MONITOR_SERIALPORT_ADDRESS	MONITOR_SERIALPORT_ADDRESS
PERFORMANCE_RECORDER_GENERATION_DIR	MONITOR_CANDERA_PERFORMANCE_RECORDER_GENERATION_DIR
PERFORMANCE_RECORDER_TEMPLATE_DIR	MONITOR_CANDERA_PERFORMANCE_RECORDER_TEMPLATE_DIR

Existing monitor settings have been moved from code to CMake settings:

> These flags are marked as advanced and should only be changed if the current settings have side effects.

Name	Description	Possible values
MONITOR_CANDERA_PERF_RECORD_BUFFER_SIZE	Defines the buffer size on target which is used for performance log tracing. When only perf-log streaming is used, it can use a smaller buffer. 4000 as size can be enough then. The effective minimum requirement depends on the amount of recorders.	positive integer
MONITOR_CANDERA_PERF_RECORD_STRID_BUFFER_SIZE	Defines the maximum of stored strings that can be sent (performance logs) with an id to reduce payload.	positive integer
MONITOR_CANDERA_PERF_RECORD_STRING_SIZE	Defines the maximum length of a single string that can be sent (performance logs).	positive integer

FeatStd Monitor

Non-blocking connection setup

The monitor supports non-blocking connection setup now. CGIAnalyzer and target can be connected any time now.

Performance logs: Payload reduction

The payload of performance logs are reduced to approx. 36 percentage of previous version. The buffer for streaming can be smaller now due to restructuring the sent data. Using the recording only functionality can store more recordings within the same buffer. The amount of recordings traced is defined by the size of buffer. Changes to the buffer size can be done by changing the CMake flag MONITOR_CANDERA_PERF_RECORD_BUFFER_SIZE.

See also:

FeatStd CMake Switches for the changed CMake feature flag related to Monitor functionality.

FeatStd Platform

64 Bit Support

Support for building with 64 bit has been added. In order to obtain a 64 bit build, one has to configure the build to use a 64 bit compiler. Doing this will automatically enable the CMake variable FEATSTD_64BIT_PLATFORM which is further used in FeatStd. If a 32 bit compiler is used, the CMake variable FEATSTD_32BIT_PLATFORM will be enabled automatically.

Two new types have been introduced for unsigned and signed integers which extend to the correct size when used in a 64 bit system or a 32 bit system. These are **FeatStd::SizeType** and **FeatStd::OffsetType** (equivalent to std::size_t type and std::ptrdiff_t types).

> The API has been adapted at various locations to use FeatStd::SizeType and FeatStd::OffsetType instead of integer types. Application code using a previous version will be compatible due to implicit integer conversion. Please take care on compiler warnings related to type conversions, alignment casts and signed/unsigned comparisons.

FeatStd Util

Thread safe String

The FeatStd::String class has been enhanced by a method String::GetCriticalSection which can be used to protect the internal c-string from modification by other threads.

> The feature is only available when CMake flag FEATSTD_THREADSAFETY_ENABLED is enabled.
