FeatStd flags

- **FEATSTD_32BIT_PLATFORM** a 32 bit compiler is used.
- **FEATSTD_64BIT_PLATFORM** a 64 bit compiler is used.
- **FEATSTD_CUSTOM_CONFIG** adds custom config header file.
- **FEATSTD_GLERRORS_AND_EGLERRORS_ENABLED** enables glGetError and eglGetError calls.
- **FEATSTD_IPC_ENABLED** enables IPC functionality, like IPC channels, etc.
- **FEATSTD_LOGGER_MAX_BUFFER_SIZE** defines the maximum length of a single log message.
- **FEATSTD_LOG_ENABLED** enables logging functionality. If disabled all logging defines are expanded empty and all logging classes (e.g. Appender) will not be available. This means the application which uses e.g. Appender has to scope this usage with FEATSTD_LOG_ENABLED.
- **FEATSTD_MEMORYPOOL_ENABLED** enables optional component MemoryPool. MemoryPool is an optional component of FeatStd.
- **FEATSTD_MEMORYPOOL_FEATURES_DEBUG** defines MemoryPool features for debug build mode.
- **FEATSTD_MEMORYPOOL_FEATURES_MINSIZEREL** defines MemoryPool features for minimal size build mode.
- **FEATSTD_MEMORYPOOL_FEATURES_RELEASE** defines MemoryPool features for debug release mode.
- **FEATSTD_MEMORYPOOL_FEATURES_RELWITHDEBINFO** defines MemoryPool features for release with debugging info build mode.
- **FEATSTD_MONITOR_ENABLED** enables monitor module.
- **FEATSTD_STRINGBUFFER_APPENDER_CANDERATYPESIZE** defines the maximum size of one Candera type used with string buffer appender.
- **FEATSTD_STRINGBUFFER_APPENDER_ENABLED** enables the additional StringBuffer AppendObject method and the StringBufferAppender template. StringBufferAppender implementations for specific types like enum, struct or class types have to be provided by the more specific library/application code (manually or generated).
- **FEATSTD_THREADSAFETY_ENABLED** activates thread safety for various FeatStd modules like logging, etc.
- **FEATSTD_UNITTEST_FRAMEWORK_ENABLED** activates unit test framework support.
- **FEATSTD_FILE_** usually should be '__FILE__' but for some special gcc build it could be set to '__BASE_FILE__'.
- **CGIAPP_ENABLE_ADDITIONAL_BEHAVIORS** defines whether additional behaviors are enabled or not
- **FEATSTD_CMAKE_LOGGING_LEVEL** set verbosity level of FeatStd CMake system

Revision #3
Created 2023-03-07 07:28:47 UTC by Tsuyoshi.Kato
Updated 2025-04-15 10:18:34 UTC by Tsuyoshi.Kato