FeatStd 3.9

Increased stack size for Integrity threads

The stack size has been increased for Integrity threads. Every thread created by FeatStd::Internal::Thread is affected. So every Candera engine thread other than the main thread has the new stack size.

Assertion function with message

Additional assertion function has been added which prints a defined messsage when hit. (9735)
The macro FEATSTD_DEBUG_ASSERT_MESSAGE(condition, message, ...) can be found in FeatStd/Diagnostics/Debug.h.
This macro is resolved into the public interface:
bool FeatStd::Diagnostics::DebugControl::Assert(const Char* expression, const Char* filename, Int lineNr, const Char *msgFormat, ...);

Option to copy StringData

A CMake option FEATSTD_STRINGDATA_COPY_ENABLED has been added for copying the internal StringData of a FeatStd::String. When the flag is enabled (default: OFF), then instead of sharing and reference counting the internal StringData, it is copied. This might be useful in application environments, where FeatStd::String objects are modified by several threads without controlling access via FeatStd::String::GetCriticalSection().

Interface Changes

- Shared Pointer

  FeatStd::MemoryManagement::SharedPointer is an intrusive smart pointer that injects 2 functions "Release" and "Retain" into the class scope of the object to be shared. Because "Release" and "Retain" are commonly used names for other functions, this can potentially result in a naming conflict with undefined behavior. Therefore these functions were renamed to "SharedPointerIntrusiveRelease" and "SharedPointerIntrusiveRetain". Any external use of these functions must be adapted.
- Migration Guide

Description	FeatStd V3.9.0	Candera V3.9.0

SharedPointer changes	Using a SharedPointer within a class introduced "Release" and "Retain" functions.	Using a SharedPointer within a class now introduces "SharedPointerIntrusiveRelease" and "SharedPointerIntrusiveRetain" instead.

Revision #2
Created 2023-03-01 06:12:14 UTC by Kanai Tomoaki
Updated 2023-06-13 10:39:32 UTC by Kanai Tomoaki