FeatStd V3.0.2

- FeatStd CMake
- FeatStd Diagnostics
- FeatStd Util

FeatStd CMake

Location of generated files

Automatically generated configuration headers are generated into `CMAKE_CURRENT_BINARY_DIR/gensrc/FeatStd` (i.e in most cases `CMAKE_BINARY_DIR/FeatStd/gensrc/FeatStd`) instead of `CMAKE_BINARY_DIR/gensrc/FeatStd`. Automatic CGI-Studio includes paths have been adapted to accommodate this change.

Add custom config header file

New build setting FEATSTD_CUSTOM_CONFIG to specifiy a custom config header file that gets included on top of generated file `CMAKE_CURRENT_BINARY_DIR/gensrc/FeatStd/Config.h`. This allows to add or override global CGI system settings (e.g. macros).

Example:

`FEATSTD_CUSTOM_CONFIG="path/to/MyConfig.h"`

MyConfig.h

```
#define FEATSTD_DEBUG_ASSERT(condition) \
    if (!(condition)) { \
        FeatStd::Internal::Generic::assert((#condition), __FILE__, __LINE__, false); \
    }
```

FeatStd Diagnostics

Debug Macros

The following macros can get overridden for custom needs:

- FEATSTD_DEBUG_BREAK
- FEATSTD_DEBUG_ASSERT
- FEATSTD_DEBUG_FAIL
- FEATSTD_DEBUG_REENTRANCE_GUARD

If these macros are not defined in custom header file the default implementation is taken, e.g. (code as in FeatStd/Diagnostics/Debug.h)

```
#if !defined(FEATSTD_DEBUG_ASSERT)              // <-- allows to override in custom header file
    #if defined(FEATSTD_DEBUG)

        FEATSTD_LINT_MACRO_WHILEFALSE(FEATSTD_DEBUG_ASSERT)
        #define FEATSTD_DEBUG_ASSERT(condition) \
                do { \
                    if (!(condition)) { \
                        if (FeatStd::Diagnostics::DebugControl::Assert((#condition), \
                                                                       __FILE__, __LINE__)) { \
                            FEATSTD_OS_BREAK(); \
                                      } \
                                } \
                            FEATSTD_SUPPRESS_MSC_WARNING_FOR_NEXT_EXPRESSION(4127, while (false)
                        } while (false)
    #else
        #define FEATSTD_DEBUG_ASSERT(condition)
    #endif
#endif
```

See also:

Add custom config header file

FeatStd Util

UTF8 String Constructor

The String class constructor now has an optional parameter, to specify whether the length of a non-null-terminated UTF8 string with multi-byte characters is given in bytes or in code-points.
