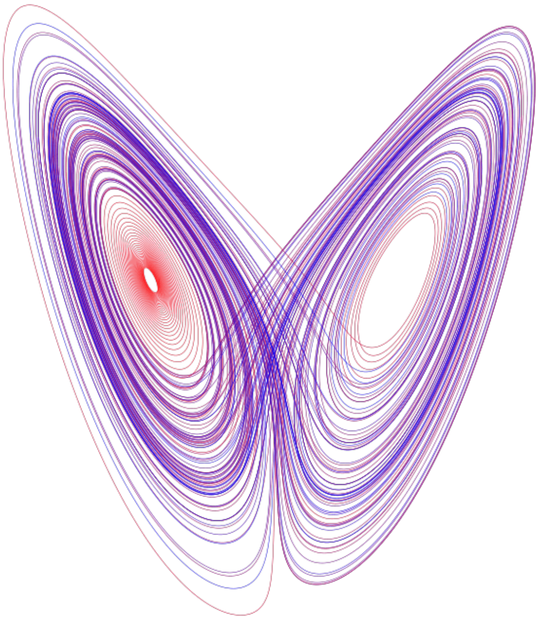

Khaos

Generated by Doxygen 1.9.2

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

Chapter 2

File Documentation

2.1 khaos/KhaosVersion.h File Reference

Macros providing the Khaos version.

Macros

- #define KHAOS_VERSION_PRIVATE()

  Khaos version.
- #define KHAOS_VERSION_TWEAK_PRIVATE()

  Khaos version tweak.

2.1.1 Detailed Description

Macros providing the Khaos version.

Copyright

2022 flagarde

Contain the macros to obtain the version and version tweak of Khaos.

Note

Use these macros only if you need preprocessor version checking.

For example :

```
#if KHAOS_GET_VERSION_MAJOR(KHAOS_VERSION == 2
  //...
#else
  //...
#endif
```

Definition in file KhaosVersion.h.

2.1.2 Macro Definition Documentation

2.1.2.1 KHAOS_VERSION_PRIVATE

```
#define KHAOS_VERSION_PRIVATE( )
```

Khaos version.

Returns

Khaos version in integer literal UL form.

Note

Use KHAOS_GET_VERSION_MAJOR(version), KHAOS_GET_VERSION_MINOR(version), KHAOS_GET_VERSION_PATCH(to parse it.

Definition at line 28 of file KhaosVersion.h.

2.1.2.2 KHAOS_VERSION_TWEAK_PRIVATE

```
#define KHAOS_VERSION_TWEAK_PRIVATE( )
```

Khaos version tweak.

Returns

Khaos version tweak in integer literal UL form.

Note

Use KHAOS_GET_VERSION_TWEAK(version) to parse it.

Definition at line 37 of file KhaosVersion.h.

2.2 KhaosVersion.h

Go to the documentation of this file.

```
00001
00019 #ifndef KHAOS_KHAOSVERSION_H_
00020 #define KHAOS_KHAOSVERSION_H_
00021
00027 // clang-format off
00028 #define KHAOS_VERSION_PRIVATE()       (((0UL % 100) * 10000000) + ((0UL % 100) * 100000) + (1UL %
       100000))
00029 // clang-format on
00030
00036 // clang-format off
00037 #define KHAOS_VERSION_TWEAK_PRIVATE() (0UL % 100000)
00038 // clang-format on
00039
00040 #endif  // KHAOS_KHAOSVERSION_H_
```

2.3 khaos/CompareVersionMacros.h File Reference

Define macros to compare versions.

```
#include "khaos/KhaosVersion.h"
#include "khaos/VersionMacros.h"
```

Macros

- #define KHAOS_VERSION_GREATER(version, major, minor, patch)

  Check if version is greater than the major, minor, patch .
- #define KHAOS_VERSION_GREATER_EQUAL(version, major, minor, patch)

  Check if version is greater or equal than major, minor, patch .
- #define KHAOS_VERSION_EQUAL(version, major, minor, patch)

  Check if version is equal than major, minor, patch .
- #define KHAOS_VERSION_LOWER_EQUAL(version, major, minor, patch)

  Check if version is lower or equal than major, minor, patch .
- #define KHAOS_VERSION_LOWER(version, major, minor, patch)

  Check if version is lower than major, minor, patch .

2.3.1 Detailed Description

Define macros to compare versions.

Copyright

2022 flagarde

Note

Use these macros only if you need preprocessor version checking.

For example :

```
#if KHAOS_VERSION_GREATER(version1,version2)
 //...
#else
 //...
#endif
```

Definition in file CompareVersionMacros.h.

2.3.2 Macro Definition Documentation

2.3.2.1 KHAOS_VERSION_EQUAL

```
#define KHAOS_VERSION_EQUAL(
                version,
                major,
                minor,
                patch )
```

Check if version is equal than major, minor, patch .

Parameters

in	*version*	.
in	*major*	.
in	*minor*	.
in	*patch*	.

Note

> Value can be directly used in both preprocessor and compiler expressions for comparison to other similarly defined values.

Definition at line 52 of file CompareVersionMacros.h.

2.3.2.2 KHAOS_VERSION_GREATER

```
#define KHAOS_VERSION_GREATER(
              version,
              major,
              minor,
              patch )
```

Check if version is greater than the major, minor, patch .

Parameters

in	*version*	.
in	*major*	.
in	*minor*	.
in	*patch*	.

Note

> Value can be directly used in both preprocessor and compiler expressions for comparison to other similarly defined values.

Definition at line 30 of file CompareVersionMacros.h.

2.3.2.3 KHAOS_VERSION_GREATER_EQUAL

```
#define KHAOS_VERSION_GREATER_EQUAL(
              version,
              major,
              minor,
              patch )
```

Check if version is greater or equal than major, minor, patch .

Parameters

in	*version*	.
in	*major*	.
in	*minor*	.
in	*patch*	.

Note

Value can be directly used in both preprocessor and compiler expressions for comparison to other similarly defined values.

Definition at line 41 of file CompareVersionMacros.h.

2.3.2.4 KHAOS_VERSION_LOWER

```
#define KHAOS_VERSION_LOWER(
              version,
              major,
              minor,
              patch )
```

Check if version is lower than major, minor, patch .

Parameters

in	*version*	.
in	*major*	.
in	*minor*	.
in	*patch*	.

Note

Value can be directly used in both preprocessor and compiler expressions for comparison to other similarly defined values.

Definition at line 74 of file CompareVersionMacros.h.

2.3.2.5 KHAOS_VERSION_LOWER_EQUAL

```
#define KHAOS_VERSION_LOWER_EQUAL(
              version,
              major,
              minor,
              patch )
```

Check if version is lower or equal than major, minor, patch .

Parameters

in	*version*	.
in	*major*	.
in	*minor*	.
in	*patch*	.

Note

Value can be directly used in both preprocessor and compiler expressions for comparison to other similarly defined values.

Definition at line 63 of file CompareVersionMacros.h.

2.4 CompareVersionMacros.h

Go to the documentation of this file.

```
00001 #ifndef KHAOS_COMPAREVERSIONMACROS_H_
00002 #define KHAOS_COMPAREVERSIONMACROS_H_
00003
00004 #include "khaos/KhaosVersion.h"
00005 #include "khaos/VersionMacros.h"
00006
00030 #define KHAOS_VERSION_GREATER(version, major, minor, patch) (version##_PRIVATE() >
       KHAOS_SET_VERSION(major, minor, patch))
00031
00041 #define KHAOS_VERSION_GREATER_EQUAL(version, major, minor, patch) (version##_PRIVATE() >=
       KHAOS_SET_VERSION(major, minor, patch))
00042
00052 #define KHAOS_VERSION_EQUAL(version, major, minor, patch) (version##_PRIVATE() ==
       KHAOS_SET_VERSION(major, minor, patch))
00053
00063 #define KHAOS_VERSION_LOWER_EQUAL(version, major, minor, patch) (version##_PRIVATE() <=
       KHAOS_SET_VERSION(major, minor, patch))
00064
00074 #define KHAOS_VERSION_LOWER(version, major, minor, patch) (version##_PRIVATE() <
       KHAOS_SET_VERSION(major, minor, patch))
00075
00076 #endif  // KHAOS_COMPAREVERSIONMACROS_H_
```

2.5 khaos/VersionMacros.h File Reference

Define macros to set, get versions.

```
#include "khaos/CompareVersionMacros.h"
#include "khaos/KhaosVersion.h"
```

Macros

- #define KHAOS_SET_VERSION(major, minor, patch)

  Set standard version numbers.
- #define KHAOS_SET_VERSION_TWEAK(tweak)

  Set standard tweak (build) numbers.
- #define KHAOS_GET_VERSION_MAJOR(version)

  Get standard major version numbers.
- #define KHAOS_GET_VERSION_MINOR(version)

  \bried Get standard minor version numbers.
- #define KHAOS_GET_VERSION_PATCH(version)

  Get standard patch version numbers.
- #define KHAOS_GET_VERSION_TWEAK(version)

  Get standard tweak version numbers.

2.5.1 Detailed Description

Define macros to set, get versions.

Copyright

2022 flagarde

Note

Use these macros only if you need preprocessor version checking.

For example :

```
#if KHAOS_GET_VERSION_MAJOR(version1) == KHAOS_GET_VERSION_MAJOR(version2)
 //...
#else
 //...
#endif
```

Definition in file VersionMacros.h.

2.5.2 Macro Definition Documentation

2.5.2.1 KHAOS_GET_VERSION_MAJOR

```
#define KHAOS_GET_VERSION_MAJOR(
              version )
```

Get standard major version numbers.

Parameters

in	*version*	The standard version number.

Note

Value can be directly used in both preprocessor and compiler expressions for comparison to other similarly defined values.

Definition at line 54 of file VersionMacros.h.

2.5.2.2 KHAOS_GET_VERSION_MINOR

```
#define KHAOS_GET_VERSION_MINOR(
              version )
```

\bried Get standard minor version numbers.

Parameters

in	*version*	The standard version number.

Note

Value can be directly used in both preprocessor and compiler expressions for comparison to other similarly defined values.

Definition at line 62 of file VersionMacros.h.

2.5.2.3 KHAOS_GET_VERSION_PATCH

```
#define KHAOS_GET_VERSION_PATCH(
              version )
```

Get standard patch version numbers.

Parameters

in	*version*	The standard version number.

Note

Value can be directly used in both preprocessor and compiler expressions for comparison to other similarly defined values.

Definition at line 70 of file VersionMacros.h.

2.5.2.4 KHAOS_GET_VERSION_TWEAK

```
#define KHAOS_GET_VERSION_TWEAK(
              version )
```

Get standard tweak version numbers.

Parameters

in	*version*	The standard version number.

Note

Value can be directly used in both preprocessor and compiler expressions for comparison to other similarly defined values.

Definition at line 78 of file VersionMacros.h.

2.5.2.5 KHAOS_SET_VERSION

```
#define KHAOS_SET_VERSION(
              major,
              minor,
              patch )
```

Set standard version numbers.

Parameters

in	*major*	The major number can be in the [0,99] range.
in	*minor*	The minor number can be in the [0,99] range.
in	*patch*	The patch number can be in the [0,99999] range.

Remarks

Decimal base whole numbers in the range [0,1000000000).

The number range is designed to allow for a (2,2,5) triplet; Which fits within a 32 bit value.

Note

Value can be directly used in both preprocessor and compiler expressions for comparison to other similarly defined values.

Values can be specified in any base. As the defined value is an constant expression.

The implementation enforces the individual ranges for the major, minor, and patch numbers.

Warning

Values over the ranges are truncated (modulo).

Definition at line 35 of file VersionMacros.h.

2.5.2.6 KHAOS_SET_VERSION_TWEAK

```
#define KHAOS_SET_VERSION_TWEAK(
              tweak )
```

Set standard tweak (build) numbers.

Parameters

in	*tweak*	The tweak number can be in the [0,9999] range.

Note

Value can be directly used in both preprocessor and compiler expressions for comparison to other similarly defined values.

Values can be specified in any base. As the defined value is an constant expression.

The implementation enforces the individual ranges for the major, minor, and patch numbers.

Warning

Values over the ranges are truncated (modulo).

Definition at line 46 of file VersionMacros.h.

2.6 VersionMacros.h

Go to the documentation of this file.

```
00001 #ifndef KHAOS_VERSIONMACROS_H_
00002 #define KHAOS_VERSIONMACROS_H_
00003
00004 #include "khaos/CompareVersionMacros.h"
00005 #include "khaos/KhaosVersion.h"
00006
00035 #define KHAOS_SET_VERSION(major, minor, patch) ((((major)*1UL % 100) * 10000000) + (((minor)*1UL %
       100) * 100000) + ((patch)*1UL % 100000))
00036
00046 #define KHAOS_SET_VERSION_TWEAK(tweak) ((tweak)*1UL % 100000)
00047
00054 #define KHAOS_GET_VERSION_MAJOR(version) (((version##_PRIVATE()) * 1UL / 10000000) % 100)
00055
00062 #define KHAOS_GET_VERSION_MINOR(version) (((version##_PRIVATE()) * 1UL / 100000) % 100)
00063
00070 #define KHAOS_GET_VERSION_PATCH(version) ((version##_PRIVATE()) * 1UL % 100000)
00071
00078 #define KHAOS_GET_VERSION_TWEAK(version) ((version##_TWEAK_PRIVATE()) * 1UL % 100000)
00079
00080 #endif  // KHAOS_VERSIONMACROS_H_
```