IO

tango.io.Conduit
- read (void[] dst) :: uint
- fill (void[] dst) :: Conduit
- write (void[] src) :: uint
- flush (void[] src) :: Conduit
- copy (IConduit src) :: Conduit
- bufferSize () :: uint
- isReadable () :: bool
- isWritable () :: bool
- isAlive () :: bool
- close () :: Conduit

tango.io.Buffer
- slice () :: void[]
- slice (uint size, bool eat) :: void[]
- append (void* content, uint length) :: Buffer
- append (void[] content) :: Buffer
- append (IBuffer other) :: Buffer
- read (void[] dst) :: uint
- readExact (void[] dst) :: Buffer
- truncate (uint extent) :: Buffer
- compress () :: Buffer
- clear () :: Buffer
- readable () :: uint
- writable () :: uint
- limit () :: uint
- capacity () :: uint
- position () :: uint
- conduit () :: Conduit
- skip (int bytes) :: Buffer
- next (uint delegate (void[])) :: bool
- fill (IConduit src) :: Buffer
- drain (IConduit dst) :: Buffer
- flush (IConduit dst) :: Buffer
- copy (IConduit src) :: Buffer
- setConduit (IConduit conduit) :: Buffer
- setContent (void[] data) :: Buffer
- setContent (void[] data, uint readable) :: Buffer
- getContent () :: void[]

File

tango.io.File
- path () :: FilePath
- read () :: void[]
- write (void[] content) :: File
- append (void[] content) :: File

tango.io.FileConduit :: Conduit
- path () :: FilePath
- length () :: ulong
- position () :: ulong
- truncate () :: FileConduit
- seek (ulong offset, Seek.Anchor anchor) :: ulong

tango.io.FilePath
- toUtf8 () :: char[]
- root ():: char[]
- folder ():: char[]
- parent ():: char[]
- name ():: char[]
- ext ():: char[]
- suffix ():: char[]
- path ():: char[]
- file ():: char[]
- set (char[] path) :: FilePath
- root (char[] other) :: FilePath
- folder (char[] other) :: FilePath
- name (char[] other) :: FilePath
- suffix (char[] other) :: FilePath
- path (char[] other) :: FilePath
- file (char[] other) :: FilePath
- join (char[][] paths…) :: void
- append (char[][] others…) :: FilePath
- prepend (char[] other) :: FilePath
- cString () :: char[]
- normalize () :: FilePath
- isAbsolute () :: bool
- isEmpty () :: bool
- isChild () :: bool
- timeStamps () :: Stamps
- modified () :: Time
- accessed () :: Time
- created () :: Time
- filesize () :: ulong
- isFolder () :: bool
- isWritable () :: bool
- create () :: FilePath
- createFile () :: FilePath
- createFolder () :: FilePath
- remove () :: FilePath
- copy (char[] src) :: FilePath
- rename (char[] dst) :: FilePath
- toList (bool prefixed) :: char[][]
- toList (void delegate (char[], char[], bool)) :: FilePath

Net

tango.net.SocketConduit :: Conduit
- socket () :: Socket
- setTimeout (Interval interval) :: SocketConduit
- connect (Address addr) :: SocketConduit
- bind (Address addr) :: SocketConduit
- shutdown ():: SocketConduit
- hadTimeout ():: bool

tango.net.DatagramConduit :: SocketConduit
- read (void[] dst, Address from) :: uint
- write (void[] src, Address to) :: uint

tango.net.MulticastConduit :: DatagramConduit
- loopback (bool yes) :: MulticastConduit
- join () :: MulticastConduit
- leave () :: MulticastConduit

tango.net.ServerSocket
- setLingerPeriod (int period) :: ServerSocket
- isAlive () :: bool
- socket () :: Socket
- accept () :: SocketConduit

tango.net.Uri
- getDefaultPort () :: uint
- getScheme () :: char[]
- getHost () :: char[]
- getPort () :: char[]
- getValidPort () :: uint
- getUserInfo () :: char[]
- getPath () :: char[]
- getQuery () :: char[]
- getFragment () :: char[]
- isGeneric () :: bool
- toUtf8 () :: char[]
- parse (char[] uri, bool relative) :: Uri
- reset () :: Uri
- relParse (char[] uri) :: Uri
- setScheme (char[] scheme) :: Uri
- setHost (char[] host) :: Uri
- setPort (int port) :: Uri
- setUserInfo (char[] info) :: Uri
- setQuery (char[] query) :: Uri
- setPath (char[] path) :: Uri
- setFragment (char[] path) :: Uri

Text

tango.text.Util

- trim (T[] src) :: T[]
- strip (T[] src, T match) :: T[]
- delimit (T[] src, T[] set) :: T[]
- split (T[] src, T[] pattern) :: T[]
- splitLines (T[] src) :: T[]
- join (T[][] src, T[] postfix, T[] output) :: T[]
- replace (T[] src, T match, T sub) :: T[]
- substitute (T[] src, T[] match, T[] sub) :: T[]
- contains (T[] src,T match) :: bool
- containsPattern (T[] src, T[] match) :: bool
- locate (T[] src,T match, int start) :: uint
- locatePrior (T[] src, T match, int start) :: uint
- locatePattern (T[] src, T[] match, int start) :: uint
- locatePatternPrior (T[] src, T[] match, int start) :: uint
- isSpace (T char) :: bool
- layout (T[] destination, T[] format ...) :: T[]
- lines (T[] str) :: LineFreach
- quotes (T[] str, T[] set) :: QuoteFreach
- delimiters (T[] str, T[] set) :: DelimFreach
- patterns (T[] str, T[] pattern, T[] sub) :: PatternFreach

tango.text.convert.Integer

- toInt (T[] src, uint radix) :: int
- toLong (T[] src, uint radix) :: long
- parse (T[] src, uint radix, uint* ate) :: long
- toUtf8 (long v) :: char[]
- toUtf16 (long v) :: wchar[]
- toUtf32 (long v) :: dchar[]
- format (T[] dst, long v, Style style, Flags flags) :: T[]

tango.text.convert.Float

- toFloat (T[] digits) :: real
- parse (T[] src, uint* ate) :: real
- toUtf8 (real v, uint decimals, bool e) :: char[]
- toUtf16 (real v, uint decimals, bool e) :: wchar[]
- toUtf32 (real v, uint decimals, bool e) :: dchar[]
- format (T[] dst, real v, uint decimals, bool e) :: T[]

tango.text.convert.Layout

- sprint (T[] result, T[] format, ...) :: T[]
- convert (T[] format, ...) :: T[]
- convert (Sink sink, T[] format, ...) :: uint

Stdio

tango.io.Print

- format (T[] fmt, ...) :: Print
- formatln (T[] fmt, ...) :: Print
- print (…) :: Print
- newline () :: Print
- flush () :: Print
- buffer () :: Buffer
- conduit () :: Conduit
- layout () :: Layout
- layout (Layout layout) :: Print

tango.io.Console.Ouput

- append (char[] content) :: Output
- append (Object object) :: Output
- newline () :: Output
- flush () :: Output
- buffer () :: Buffer
- conduit () :: Conduit
- redircted () :: bool

tango.io.Console.Input

- copyln (bool raw) :: char[]
- readln (inout char[] line, bool raw) :: bool
- buffer () :: Buffer
- conduit () :: Conduit
- redircted () :: bool

tango.io.Console

- Cin :: Input
- Cout :: Output
- Cerr :: Output

tango.io.Stdout

- Stdout :: Print
- Stderr :: Print

Utils

tango.util.time.Utc

- time () :: Time
- zone () :: int
- local () :: Time
- toLocal (Time time) :: Time
- fromLocal (Time time) :: Time

tango.util.time.Date

- setDate (int year, int month, int day, int dow) :: void
- setTime (int hour, int min, int sec, int ms = 0) :: void
- set (Time time) :: void
- get () :: Time

year	fully defined year ~ e.g. 2005
month	1 through 12
day	1 through 31
hour	0 through 23
min	0 through 59
sec	0 through 59
ms	0 through 999
dow	0 through 6; sunday == 0

tango.util.log.Log

- getLogger (char[] name) :: Logger

tango.util.log.Logger

- trace (lazy char[] exp) :: Logger
- info (lazy char[] exp) :: Logger
- warn (lazy char[] exp) :: Logger
- error (lazy char[] exp) :: Logger
- fatal (lazy char[] exp) :: Logger
- name () :: char[]
- level () :: Logger
- setLevel (Level level) :: Logger
- isEnabled (Level level) :: bool
- addAppender (Appender app) :: Logger
- clearAppenders () :: Logger
- runtime () :: Time

Math

tango.math.Math

- abs (T value) :: T
- minNum (real x, real y) :: real
- maxNum (real x, real y) :: real
- cos (real x) :: real
- sin (real x) :: real
- tan (real x) :: real
- acos (real x) :: real
- asin (real x) :: real
- atan (real x) :: real
- atan2 (real x) :: real
- cosh (real x) :: real
- sinh (real x) :: real
- tanh (real x) :: real
- acosh (real x) :: real
- asinh (real x) :: real
- atanh (real x) :: real
- cosPi (real x) :: real
- sinPi (real x) :: real
- atanPi (real x) :: real
- sqrt (real x) :: real
- cbrt (real x) :: real
- exp (real x) :: real
- expm1 (real x) :: real
- exp2 (real x) :: real
- log (real x) :: real
- log1p (real x) :: real
- log2 (real x) :: real
- log10 (real x) :: real
- pow (real x, uint n) :: real
- pow (real x, real y) :: real
- hypot (real x, real y) :: real
- poly (real x, real[] coeff) :: real
- rationalPoly (real x, real[] numer, real[] denom) :: real
- floor (real x) :: real
- ceil (real x) :: real
- round (real x) :: real
- trunc (real x) :: real
- rndint (real x) :: int
- rndlong (real x) :: long

String

tango.text.String

- set (T[] content) :: String
- set (StringView other) :: String
- selection () :: T[]
- selectionSpan () :: Span
- select (int start, int length) :: String
- select (T c) :: bool
- select (T[] pattern) :: bool
- select (StringView other) :: bool
- selectPrior (T c) :: bool
- selectPrior (T[] pattern) :: bool
- selectPrior (StringView other) :: bool
- append (StringView other) :: String
- append (T[] content) :: String
- append (T char, int count) :: String
- append (int value) :: String
- append (long value) :: String
- append (real value) :: String
- encode (char[] content) :: String
- encode (wchar[] content) :: String
- encode (dchar[] content) :: String
- prepend (T[] content) :: String
- prepend (StringView other) :: String
- prepend (T char, int count) :: String
- replace (T char) :: String
- replace (T[] content) :: String
- replace (StringView other) :: String
- remove () :: String
- clear () :: String
- trim () :: String
- strip (T char) :: String
- truncate (int point) :: String
- reserve (int extra) :: String
- toHash () :: uint
- length () :: uint
- equals (T[] text) :: bool
- equals (StringView other) :: bool
- ends (T[] text) :: bool
- ends (StringView other) :: bool
- starts (T[] text) :: bool
- starts (StringView other) :: bool
- compare (T[] text) :: int
- compare (StringView other) :: int
- copy (T[] dst) :: T[]
- slice () :: T[]
- encoding () :: typeinfo
- comparator (Comparator other) :: Comparator
- toUtf8 (char[] dst) :: char[]
- toUtf16 (wchar[] dst) :: wchar[]
- toUtf32 (dchar[] dst) :: dchar[]