

NitroPascal Language Coverage

This document tracks all language features planned for NitroPascal. Each feature is marked as implemented [x] or planned []. This serves as the master TODO list for the project.

INSTRUCTIONS FOR UPDATES:

ONLY add ticks ☒ to mark features as implemented

- Find the appropriate section for the feature and tick it there
- If no appropriate section exists, create a new section with the proper feature list and tick items there
- DO NOT add any other text, commentary, or explanations
- Keep all content as tick lists only

Basic Types

- ☒ Integer → int32_t
- ☒ Int64 → int64_t
- ☒ Cardinal → uint32_t
- ☒ Byte → uint8_t
- ☒ Word → uint16_t
- ☒ Boolean → bool
- ☒ Double → double
- ☒ Single → float
- ☒ Char → char16_t (UTF-16)
- ☒ String → np::String (UTF-16)
- ☒ Pointer → T*
- ☒ Untyped Pointer → void*
- ☐ ShortInt → int8_t
- ☐ SmallInt → int16_t
- ☐ LongInt → int32_t
- ☐ LongWord → uint32_t
- ☐ UInt64 → uint64_t
- ☐ Extended → long double
- ☐ Currency → custom type
- ☐ Real → double
- ☐ Real48 → custom type
- ☐ AnsiString → std::string
- ☐ WideString → std::wstring
- ☐ UnicodeString → std::u16string
- ☐ AnsiChar → char
- ☐ WideChar → wchar_t

Type System

- ☒ Type aliases (type T = OtherType)
- ☒ Pointer type declarations (^T)
- ☒ Pointer type aliases
- ☐ Subrange types (type TDigit = 0..9)
- ☐ Type casting (Integer(x))
- ☐ Type conversion operators
- ☐ Type compatibility checking
- ☐ Ordinal types
- ☐ Type identity vs assignment compatibility

Constants

- ☒ Const declarations
- ☒ Typed constants
- ☒ Integer constants
- ☒ Float constants
- ☒ String constants
- ☒ Boolean constants
- ☐ Enumeration constants
- ☐ Set constants
- ☐ Array constants
- ☐ Record constants
- ☐ Pointer constants (nil)
- ☐ Constant expressions
- ☐ Resource string constants

Variables

- ☒ Variable declarations (var)
- ☒ Local variables
- ☒ Global variables
- ☒ Multiple variables same type
- ☐ Variable initialization
- ☐ Thread-local variables (threadvar)
- ☐ Absolute variables (var X: Integer absolute Y)
- ☐ External variables (external)

Operators - Arithmetic

- ☒ Addition (+)
- ☒ Subtraction (-)
- ☒ Multiplication (*)
- ☒ Float division (/)
- ☒ Integer division (div)
- ☒ Modulo (mod)
- ☐ Unary plus (+X)
- ☐ Unary minus (-X)

Operators - Comparison

- ☒ Equal (=)
- ☒ Not equal (<>)
- ☒ Less than (<)
- ☒ Greater than (>)
- ☒ Less than or equal (<=)
- ☒ Greater than or equal (>=)

Operators - Logical

- ☒ Logical AND (and)
- ☒ Logical OR (or)
- ☒ Logical XOR (xor)
- ☒ Logical NOT (not)

Operators - Bitwise

- ☒ Bitwise AND (and)
- ☒ Bitwise OR (or)
- ☒ Bitwise XOR (xor)
- ☒ Bitwise NOT (not)
- ☒ Shift left (shl)
- ☒ Shift right (shr)

Operators - Assignment

- ☒ Simple assignment (:=)
- ☐ Addition assignment (+=)
- ☐ Subtraction assignment (-=)
- ☐ Multiplication assignment (*=)
- ☐ Division assignment (/=)

Operators - Other

- ☒ String concatenation (+)
- ☒ Pointer dereferencing (^)
- ☒ Address-of operator (@)
- ☒ Set membership (in)
- ☐ Range operator (..)
- ☐ Type cast operator (as)
- ☐ Type check operator (is)

Control Flow - Conditional

- ☒ if..then
- ☒ if..then..else
- ☒ Nested if statements
- ☒ case..of..end (integer)
- ☒ case..of..end (enumeration)
- ☐ case..of..end (string)
- ☐ case..of..end (ranges)
- ☐ case with multiple values per branch

Control Flow - Loops

- ☒ for..to..do
- ☒ for..downto..do
- ☒ while..do
- ☒ repeat..until
- ☐ for..in..do (arrays)
- ☐ for..in..do (collections)
- ☒ break
- ☒ continue
- ☒ exit
- ☒ exit with return value

Control Flow - Structured

- ☒ with statement
- ☒ try..except
- ☒ try..finally
- ☒ try..except..finally
- ☒ raise statement
- ☒ on E: Exception do
- ☐ else in except blocks

Functions & Procedures

- ☒ Function declarations
- ☒ Procedure declarations
- ☒ Function return values (Result)
- ☒ Parameter passing - by value
- ☒ Parameter passing - const
- ☒ Parameter passing - var
- ☒ Parameter passing - out
- ☒ Local variables
- ☒ External declarations (external 'library.dll')
- ☐ Default parameter values
- ☐ Named parameters
- ☐ Parameter arrays (array of const)
- ☐ Open array parameters
- ☐ Function overloading
- ☒ Forward declarations
- ☐ Inline functions
- ☐ Nested functions
- ☐ Anonymous functions
- ☐ Function pointers (procedural types)
- ☐ Function return in parameter (out)

Arrays - Static

- ☒ Static array declarations
- ☒ Array indexing
- ☒ Multi-dimensional arrays
- ☒ Array bounds checking
- ☐ Array initialization
- ☐ Array of const
- ☐ Zero-based arrays ([0..N])
- ☐ Custom-based arrays ([Low..High])
- ☐ Array assignment
- ☐ Array comparison
- ☐ Array slicing

Arrays - Dynamic

- ☒ Dynamic array declarations (array of T)
- ☒ SetLength for dynamic arrays
- ☒ Length for dynamic arrays
- ☒ Copy for dynamic arrays
- ☐ Dynamic array assignment
- ☐ Dynamic array initialization
- ☐ Multi-dimensional dynamic arrays
- ☐ Array concatenation
- ☐ Low function for dynamic arrays
- ☒ High function for dynamic arrays

Arrays - Built-in Functions

- ☒ Length(array)
- ☒ Low(array)
- ☒ High(array)
- ☒ SetLength(array, length)
- ☒ Copy(array, start, count)

Records

- ☒ Record type declarations
- ☒ Record field access
- ☒ Nested records
- ☒ Pointer-to-record types
- ☐ Record initialization
- ☐ Record assignment
- ☐ Record comparison
- ☐ Packed records
- ☐ Aligned records
- ☐ Variant records
- ☐ Records with methods
- ☐ Record constructors
- ☐ Record operators
- ☐ Record helpers
- ☐ Record case statements
- ☐ Record class operators

Enumerations

- ☒ Enum type declarations
- ☒ Enum values
- ☒ Enum in case statements
- ☐ Enum with explicit values
- ☐ Enum subranges
- ☐ Scoped enumerations
- ☐ Ord function for enums
- ☐ Succ function for enums
- ☐ Pred function for enums

Sets

- ☒ Set type declarations (set of)
- ☐ Set literals
- ☐ Set operations (union +)
- ☐ Set operations (difference -)
- ☐ Set operations (intersection *)
- ☐ Set operations (symmetric difference > <)
- ☒ Set membership (in)
- ☐ Set comparison (=, < >)
- ☐ Set subset (<=)
- ☐ Set superset (>=)
- ☒ Include procedure
- ☒ Exclude procedure

Pointers

- ☒ Pointer type declarations (^T)
- ☒ New procedure
- ☒ Dispose procedure
- ☒ Pointer dereferencing (P^)
- ☒ Address-of operator (@)
- ☒ nil constant → nullptr
- ☒ Typed pointers
- ☐ Pointer arithmetic
- ☐ PChar type
- ☐ Pointer comparison
- ☒ GetMem function
- ☒ FreeMem procedure
- ☒ ReallocMem function
- ☐ AllocMem function

Classes

- ☐ Class declarations
- ☐ Class fields (private, public, protected)
- ☐ Class methods
- ☐ Class properties
- ☐ Constructors (Create)
- ☐ Destructors (Destroy)
- ☐ Class inheritance
- ☐ Virtual methods
- ☐ Abstract methods
- ☐ Override directive
- ☐ Reintroduce directive
- ☐ Class methods (class function/procedure)
- ☐ Class properties
- ☐ Class variables
- ☐ Class constructors/destructors
- ☐ Self keyword
- ☐ Inherited keyword
- ☐ Class references (TClass)
- ☐ Class.ClassName
- ☐ Class.ClassType
- ☐ Object is TClass
- ☐ Object as TClass
- ☐ Class helpers
- ☐ Class operators

Interfaces

- ☐ Interface declarations
- ☐ Interface implementation (implements)
- ☐ Interface inheritance
- ☐ Interface methods
- ☐ Interface properties
- ☐ GUID support
- ☐ IUnknown/IInterface
- ☐ Reference counting
- ☐ Supports operator
- ☐ Interface delegation
- ☐ Interface method resolution

Properties

- ☐ Property declarations
- ☐ Property getters
- ☐ Property setters
- ☐ Array properties
- ☐ Indexed properties
- ☐ Default properties
- ☐ Read-only properties
- ☐ Write-only properties
- ☐ Class properties
- ☐ Property directives (stored, default)

Generics

- ☐ Generic types (type T)
- ☐ Generic constraints
- ☐ Generic methods
- ☐ Generic type parameters
- ☐ Multiple generic parameters
- ☐ Generic arrays
- ☐ Generic records
- ☐ Generic classes
- ☐ Generic interfaces
- ☐ Default function for generics

String Functions

- ☒ Length(S)
- ☒ String concatenation (+)
- ☒ String indexing (S[i], 1-based)
- ☒ Copy(S, Index, Count)
- ☒ Pos(Substr, S)
- ☒ Delete(S, Index, Count)
- ☒ Insert(Substr, S, Index)
- ☒ IntToStr(I)
- ☒ StrToInt(S)
- ☒ StrToIntDef(S, Default)
- ☒ FloatToStr(F)
- ☒ StrToFloat(S)
- ☒ Format(Fmt, Args)
- ☒ UpperCase(S)
- ☒ LowerCase(S)
- ☒ Trim(S)
- ☒ TrimLeft(S)
- ☒ TrimRight(S)
- ☒ StringReplace(S, Old, New, Flags)
- ☒ CompareStr(S1, S2)
- ☐ CompareText(S1, S2)
- ☐ SameStr(S1, S2)
- ☒ SameText(S1, S2)
- ☐ AnsiUpperCase(S)
- ☐ AnsiLowerCase(S)
- ☐ AnsiCompareStr(S1, S2)
- ☐ AnsiCompareText(S1, S2)
- ☒ QuotedStr(S)
- ☐ AnsiQuotedStr(S, Quote)
- ☐ StringOfChar(Ch, Count)
- ☐ WrapText(S, MaxCol)

Math Functions

- ☒ Abs(X)
- ☒ Sqr(X)
- ☒ Sqrt(X)
- ☒ Sin(X)
- ☒ Cos(X)
- ☒ Tan(X)
- ☒ ArcSin(X)
- ☒ ArcCos(X)
- ☒ ArcTan(X)
- ☐ Ln(X)
- ☐ Exp(X)
- ☐ Power(Base, Exponent)
- ☒ Round(X)
- ☒ Trunc(X)
- ☐ Int(X)
- ☐ Frac(X)
- ☒ Ceil(X)
- ☒ Floor(X)
- ☐ Pi constant
- ☒ Max(A, B)
- ☒ Min(A, B)
- ☒ Random
- ☒ Randomize
- ☐ RandomRange(Min, Max)

Type Conversion Functions

- ☒ IntToStr(I)
- ☒ StrToInt(S)
- ☒ StrToIntDef(S, Default)
- ☒ FloatToStr(F)
- ☒ StrToFloat(S)
- ☒ BoolToStr(B)
- ☐ StrToBool(S)
- ☒ Chr(I)
- ☒ Ord(Ch)

Ordinal Functions

- ☒ Ord(X)
- ☒ Succ(X)
- ☒ Pred(X)
- ☒ Inc(X)
- ☒ Inc(X, N)
- ☒ Dec(X)
- ☒ Dec(X, N)
- ☒ Low(X)
- ☒ High(X)

Memory Functions

- ☒ New(P)
- ☒ Dispose(P)
- ☒ GetMem(P, Size)
- ☒ FreeMem(P)
- ☒ ReallocMem(P, Size)
- ☐ AllocMem(Size)
- ☒ FillChar(Dest, Count, Value)
- ☒ Move(Source, Dest, Count)
- ☐ FillByte(Dest, Count, Value)
- ☐ MoveChars(Source, Dest, Count)

I/O Functions - Console

- ☒ Write (basic)
- ☒ WriteLn (basic)
- ☐ Write (formatted)
- ☐ WriteLn (formatted)
- ☐ Read(Var)
- ☒ ReadLn(Var)
- ☐ Eof
- ☐ Eoln
- ☐ ClrScr
- ☐ GotoXY(X, Y)
- ☐ WhereX
- ☐ WhereY
- ☐ TextColor(Color)
- ☐ TextBackground(Color)

I/O Functions - Files

- ☒ Assign(F, Name)
- ☒ Reset(F)
- ☒ Rewrite(F)
- ☒ Append(F)
- ☒ Close(F)
- ☐ Read(F, Var)
- ☒ ReadLn(F, Var)
- ☒ Write(F, Data)
- ☒ WriteLn(F, Data)
- ☒ BlockRead(F, Buf, Count)
- ☒ BlockWrite(F, Buf, Count)
- ☒ Seek(F, Position)
- ☒ FilePos(F)
- ☒ FileSize(F)
- ☒ Eof(F)
- ☐ Eoln(F)
- ☐ Erase(F)
- ☐ Rename(F, NewName)
- ☐ IOResult

File Management

- ☒ FileExists(FileName)
- ☒ DirectoryExists(DirName)
- ☒ CreateDir(DirName)
- ☐ RemoveDir(DirName)
- ☒ GetCurrentDir
- ☐ SetCurrentDir(DirName)
- ☒ DeleteFile(FileName)
- ☒ RenameFile(OldName, NewName)
- ☐ ChangeFileExt(FileName, Extension)
- ☐ ExtractFilePath(FileName)
- ☐ ExtractFileName(FileName)
- ☐ ExtractFileExt(FileName)
- ☐ ExtractFileDir(FileName)
- ☐ ExtractFileDrive(FileName)
- ☐ ExpandFileName(FileName)
- ☐ FileAge(FileName)
- ☐ FileGetAttr(FileName)
- ☐ FileSetAttr(FileName, Attr)

Date/Time Functions

- ☐ Now
- ☐ Date
- ☐ Time
- ☐ DateToStr(Date)
- ☐ TimeToStr(Time)
- ☐ DateTimeToStr(DateTime)
- ☐ StrToDate(S)
- ☐ StrToTime(S)
- ☐ StrToDateTime(S)
- ☐ FormatDateTime(Format, DateTime)
- ☐ EncodeDate(Year, Month, Day)
- ☐ EncodeTime(Hour, Min, Sec, MSec)
- ☐ DecodeDate(Date, Year, Month, Day)
- ☐ DecodeTime(Time, Hour, Min, Sec, MSec)
- ☐ DayOfWeek(Date)
- ☐ IsLeapYear(Year)
- ☐ IncMonth(Date, Months)

Compiler Directives - Build Settings

- ☒ {\$optimization mode}
- ☒ {\$target triple}
- ☒ {\$exceptions on|off}
- ☒ {\$strip on|off}
- ☒ {\$include_path path}
- ☒ {\$library_path path}
- ☒ {\$link library}
- ☒ {\$module_path path}
- ☒ {\$apptype console|gui}
- ☐ {\$define symbol}
- ☐ {\$undef symbol}
- ☒ {\$ifdef symbol}
- ☐ {\$ifndef symbol}
- ☐ {\$if expression}
- ☒ {\$else}
- ☐ {\$elseif expression}
- ☒ {\$endif}
- ☐ {\$ifopt switch}

Compiler Directives - Code Control

- ☐ {\$inline on|off}
- ☐ {\$rangechecks on|off}
- ☐ {\$overflowchecks on|off}
- ☐ {\$iochecks on|off}
- ☐ {\$typedaddress on|off}
- ☐ {\$writeableconst on|off}
- ☐ {\$booleval on|off}
- ☐ {\$assertions on|off}
- ☐ {\$optimization on|off}
- ☐ {\$stackframes on|off}
- ☐ {\$hints on|off}
- ☐ {\$warnings on|off}

Compiler Directives - Include

- ☐ {\$I filename} or {\$include filename}
- ☐ {\$include_once filename}
- ☐ {\$resource filename}

Units & Modules

- ☒ Unit declarations
- ☒ Unit interface section
- ☒ Unit implementation section
- ☒ Uses clause
- ☒ Unit namespaces
- ☐ Unit initialization section
- ☐ Unit finalization section
- ☐ Circular unit references
- ☐ Unit aliases
- ☐ Unit versioning

Program Structure

- ☒ Program declarations
- ☒ Library declarations
- ☒ Package declarations
- ☐ Uses clause in program
- ☒ Program parameters (ParamCount, ParamStr)

Exception Handling

- ☐ Exception class hierarchy
- ☒ try..except blocks
- ☒ try..finally blocks
- ☒ try..except..finally blocks
- ☒ raise statement
- ☐ raise with at
- ☒ on E: Exception do
- ☐ else in except
- ☐ Exception.Message
- ☐ Exception.ClassName
- ☐ ExceptObject function
- ☐ ExceptionClass function
- ☐ AcquireExceptionObject
- ☐ ReleaseExceptionObject

Attributes

- ☐ Custom attributes
- ☐ Attribute declarations
- ☐ Attribute parameters
- ☐ RTTI for attributes
- ☐ TCustomAttribute base class

Anonymous Methods

- ☐ Anonymous method declarations
- ☐ Anonymous method types
- ☐ Anonymous method capture
- ☐ Anonymous method parameters
- ☐ Anonymous method as properties

RTTI (Run-Time Type Information)

- ☐ TypeInfo function
- ☐ TObject.ClassInfo
- ☐ TObject.ClassName
- ☐ TObject.ClassType
- ☐ TObject.InheritsFrom
- ☐ GetTypeData function
- ☐ GetEnumName function
- ☐ GetEnumValue function
- ☐ TRttiContext
- ☐ TRttiType
- ☐ TRttiMethod
- ☐ TRttiProperty
- ☐ TRttiField

Operators Overloading

- ☐ operator overload declarations
- ☐ Implicit operator
- ☐ Explicit operator
- ☐ Add operator (+)
- ☐ Subtract operator (-)
- ☐ Multiply operator (*)
- ☐ Divide operator (/)
- ☐ IntDivide operator (div)
- ☐ Modulus operator (mod)
- ☐ Equal operator (=)
- ☐ NotEqual operator (< >)
- ☐ GreaterThan operator (>)
- ☐ LessThan operator (<)
- ☐ GreaterThanOrEqual operator (>=)
- ☐ LessThanOrEqual operator (<=)
- ☐ LogicalAnd operator (and)
- ☐ LogicalOr operator (or)
- ☐ LogicalXor operator (xor)
- ☐ LogicalNot operator (not)
- ☐ Negative operator (unary -)
- ☐ Positive operator (unary +)
- ☐ Inc operator
- ☐ Dec operator
- ☐ In operator

Helpers

- ☐ Class helpers
- ☐ Record helpers
- ☐ Type helpers
- ☐ Helper for built-in types
- ☐ Helper inheritance

Advanced Language Features

- ☐ Inline assembly
- ☐ Absolute addressing (var X: Integer absolute \$0040:\$0017)
- ☐ External variables
- ☐ External functions
- ☐ Platform attribute
- ☐ Deprecated attribute
- ☐ Experimental attribute
- ☐ Library attribute

Collections - System.Generics.Collections

- ☐ TList
- ☐ TQueue
- ☐ TStack
- ☐ TDictionary<K,V>
- ☐ TObjectList
- ☐ TObjectQueue
- ☐ TObjectStack
- ☐ TObjectDictionary<K,V>
- ☐ TThreadList
- ☐ TThreadedQueue

Collections - System.Classes

- ☐ TStringList
- ☐ TList
- ☐ TCollection
- ☐ TCollectionItem
- ☐ TComponent
- ☐ TPersistent
- ☐ TStream
- ☐ TFileStream
- ☐ TMemoryStream
- ☐ TStringStream
- ☐ TBytesStream

Threading

- ☐ TThread class
- ☐ BeginThread function
- ☐ EndThread procedure
- ☐ TMonitor
- ☐ TCriticalSection
- ☐ TMultiReadExclusiveWriteSynchronizer
- ☐ TEvent
- ☐ TMutex
- ☐ TSemaphore
- ☐ TThreadPool
- ☐ Parallel.For
- ☐ TParallel.For
- ☐ TTask
- ☐ ITask
- ☐ TTask.Run
- ☐ TTask.Wait
- ☐ TTask.WaitForAll
- ☐ TTask.WaitForAny

Variants

- ☐ Variant type
- ☐ Variant creation
- ☐ Variant type conversion
- ☐ Variant operators
- ☐ VarType function
- ☐ VarIsNull function
- ☐ VarIsEmpty function
- ☐ VarIsClear function
- ☐ VarIsNumeric function
- ☐ VarIsStr function
- ☐ VarIsArray function
- ☐ VarArrayCreate function
- ☐ VarArrayOf function
- ☐ VarArrayDimCount function
- ☐ VarArrayLowBound function
- ☐ VarArrayHighBound function

COM/OLE

- ☐ Interface reference counting
- ☐ IUnknown interface
- ☐ IDispatch interface
- ☐ CoCreateInstance
- ☐ CoInitialize
- ☐ CoUninitialize
- ☐ Variant for OLE Automation
- ☐ CreateOleObject function
- ☐ GetActiveOleObject function

Platform Services

- ☒ ParamCount function
- ☒ ParamStr function
- ☐ GetEnvironmentVariable function
- ☐ SetEnvironmentVariable procedure
- ☐ ExitCode variable
- ☒ Halt procedure
- ☐ RunError procedure
- ☐ GetTickCount function
- ☐ Sleep procedure

Code Generation

- ☒ Header file (.h) generation
- ☒ Implementation file (.cpp) generation
- ☒ Namespace per unit
- ☒ Proper includes
- ☒ Forward declarations
- ☐ Comment preservation
- ☐ Pragma once guards
- ☐ Include guard defines
- ☐ Line number directives (#line)
- ☐ Source position tracking
- ☐ Debug information generation

Build System

- ☒ Program → Executable
- ☒ Library → Shared library (.dll/.so/.dylib)
- ☒ Unit → Static library (.lib/.a)
- ☒ Zig build system integration
- ☒ Cross-platform compilation
- ☒ Build optimization modes (Debug, ReleaseSafe, ReleaseFast, ReleaseSmall)
- ☒ Conditional compilation support (DEBUG/RELEASE defines)
- ☒ Platform defines (WIN32/WIN64/LINUX/MACOS/MSWINDOWS/POSIX)
- ☒ Architecture defines (CPUX64/CPU386/CPUARM64)
- ☒ Application type defines (CONSOLE_APP/GUI_APP)
- ☒ Windows subsystem control (console vs GUI)
- ☐ Incremental compilation
- ☐ Precompiled headers
- ☐ Package management
- ☐ Dependency resolution
- ☐ Build scripts
- ☐ Custom build steps

Debugging Support

- ☐ Debug symbols generation
- ☐ Source line mapping
- ☐ Variable inspection
- ☐ Breakpoint support
- ☐ Stack trace generation
- ☐ Memory leak detection
- ☐ Assert statements
- ☐ DebugBreak support

Standard Library - System

- ☒ WriteLn
- ☒ Write
- ☒ ReadLn
- ☐ Read
- ☒ Length
- ☒ SetLength
- ☒ Copy
- ☒ Delete
- ☒ Insert
- ☒ Pos
- ☒ New
- ☒ Dispose
- ☒ GetMem
- ☒ FreeMem
- ☒ Inc
- ☒ Dec
- ☒ Succ
- ☒ Pred
- ☒ Ord
- ☒ Chr
- ☒ Low
- ☒ High
- ☒ SizeOf
- ☐ TypeOf
- ☒ Assigned
- ☐ Default
- ☐ Initialize
- ☐ Finalize

Standard Library - System.SysUtils

- ☒ IntToStr
- ☒ StrToInt
- ☒ StrToIntDef
- ☒ FloatToStr
- ☒ StrToFloat
- ☒ Format
- ☐ FormatFloat
- ☐ FormatDateTime
- ☒ UpperCase
- ☒ LowerCase
- ☒ Trim
- ☒ TrimLeft
- ☒ TrimRight
- ☒ StringReplace
- ☒ QuotedStr
- ☐ ExtractFilePath
- ☐ ExtractFileName
- ☐ ExtractFileExt
- ☐ ChangeFileExt
- ☒ FileExists
- ☒ DirectoryExists
- ☒ DeleteFile
- ☒ RenameFile
- ☒ CreateDir
- ☐ RemoveDir
- ☐ Now
- ☐ Date
- ☐ Time
- ☐ EncodeDate
- ☐ EncodeTime
- ☐ DecodeDate
- ☐ DecodeTime

Standard Library - System.Math

- ☒ Abs
- ☒ Sqr
- ☒ Sqrt
- ☒ Sin
- ☒ Cos
- ☒ Tan
- ☒ ArcSin
- ☒ ArcCos
- ☒ ArcTan
- ☐ ArcTan2
- ☐ Ln
- ☐ Exp
- ☐ Power
- ☐ IntPower
- ☒ Round
- ☐ RoundTo
- ☒ Trunc
- ☐ Int
- ☐ Frac
- ☒ Ceil
- ☒ Floor
- ☒ Max
- ☒ Min
- ☐ MaxValue
- ☐ MinValue
- ☐ MaxIntValue
- ☐ MinIntValue
- ☐ InRange
- ☐ EnsureRange
- ☐ Sign
- ☐ CompareValue
- ☐ SameValue
- ☐ IsNan
- ☐ IsInfinite
- ☐ IsZero

C/C++ Interoperability

- ☒ External function declarations (external 'library.dll')
- ☒ External library tracking for linking
- ☒ Calling conventions (stdcall, cdecl, fastcall)
- ☒ Automatic string type conversion for external functions
- ☒ PChar/PWideChar → const wchar_t* mapping
- ☒ PAnsiChar → const char* mapping
- ☒ Call-site string conversion (.c_str_wide(), .to_ansi())
- ☒ Pascal type → C type mapping for external declarations
- ☒ No declaration generation for external functions (linker resolves)
- ☐ External variable declarations
- ☐ C header file import
- ☐ C++ class wrapping
- ☐ C++ template wrapping
- ☐ C array handling
- ☐ Structure alignment
- ☐ Packed structures
- ☐ Name mangling control

Tools & Utilities

- ☒ Command-line compiler (nitro)
- ☒ Build command
- ☒ Run command
- ☒ Clean command
- ☒ Init command (project creation)
- ☒ Version command
- ☒ Help command
- ☐ Convert-header command (C header conversion)
- ☐ Package manager
- ☐ Documentation generator
- ☐ Profiler
- ☐ Code formatter
- ☐ Static analyzer
- ☐ Refactoring tools
- ☐ IDE integration
- ☐ Language server protocol (LSP)
- ☐ Syntax highlighting definitions

Runtime Library (RTL)

- ☒ I/O functions (Write, WriteLn)
- ☒ Control flow helpers (ForLoop, WhileLoop, RepeatUntil)
- ☒ Basic operators (Div, Mod, Shl, Shr)
- ☒ String class (np::String)
- ☒ Dynamic arrays (np::DynArray)
- ☒ Sets (np::Set)
- ☒ Exception handling
- ☒ Memory management
- ☐ Thread support
- ☐ RTTI support
- ☐ Collection classes
- ☒ File I/O
- ☐ Date/Time support
- ☒ Math functions
- ☒ String functions
- ☒ Type conversion functions

Cross-Platform Support

- ☒ Windows (x64)
- ☒ Linux (x64)
- ☒ macOS (x64)
- ☐ Windows (ARM64)
- ☐ Linux (ARM64)
- ☐ macOS (ARM64/Apple Silicon)
- ☐ FreeBSD
- ☐ WebAssembly (WASI)
- ☐ Android
- ☐ iOS
- ☐ Embedded systems
- ☐ Raspberry Pi

Optimization

- ☒ Optimization levels (Debug, ReleaseSafe, ReleaseFast, ReleaseSmall)
- ☐ Inline functions
- ☐ Dead code elimination
- ☐ Constant folding
- ☐ Constant propagation
- ☐ Loop unrolling
- ☐ Tail call optimization
- ☐ Register allocation hints
- ☐ Profile-guided optimization
- ☐ Link-time optimization (LTO)

Testing & Quality

- ☐ Unit testing framework
- ☒ Test runner
- ☐ Code coverage analysis
- ☐ Memory leak detection
- ☐ Static analysis
- ☐ Lint warnings
- ☐ Code metrics
- ☐ Continuous integration support

Documentation

- ☒ User manual
- ☒ Design documentation
- ☒ Language coverage list
- ☐ API reference
- ☐ Tutorial series
- ☐ Cookbook/recipes
- ☐ Migration guide (Delphi → NitroPascal)
- ☐ Performance guide
- ☐ Best practices guide
- ☐ Troubleshooting guide