

# OBJECTIVE-C FOUNDATION CLASSES

## REFERENCE CARD

### Part 5: Compound classes

#### DArguments

##### Methods

- **init** ..... Init an argument parser
- **deepen** ..... Not implemented
- **free** ..... Free the parser, incl. options
- **option** : (char \*) longOpt : (char) shortOpt  
    | : (char \*) descr : (id) target ..... Add an option
- **options** : (DArgOption \*) options : (int) nr  
    | Add multiple options
- **parse** : (char \*) name : (char \*) usage : (char \*) version  
    | : (char \*) tail : (char \*\*) argv : (int) argc  
    | Parse and process the arguments
- **printVersion** : (char \*) version ..... Print version info
- **printOptionHelp** : (char) shortOpt : (int) longCol  
    | : (char \*) longOpt : (char \*) descr .... Print option help
- **printHelp** : (char \*) usage : (char \*) tail ... Print all help

#### DCalendar

##### Constants

- DCL\_SUNDAY ..... Sunday
- DCL\_MONDAY ..... Monday
- DCL\_TUESDAY ..... Tuesday
- DCL\_WEDNESDAY ..... Wednesday
- DCL\_THURSDAY ..... Thursday
- DCL\_FRIDAY ..... Friday
- DCL\_SATURDAY ..... Saturday
- DCL\_ALL\_MONTHS ..... All months in calendar

##### Methods

- + (int) **firstWeekDay** ..... Return first day of week
- + **firstWeekDay** (int) day ..... Set first day of week
- + (BOOL) **isLeapYear** : (int) year ..... Test for leap year
- + (int) **leapYears** : (int) from : (int) to ... Count leap years
- + (int) **daysInMonth** : (int) year : (int) month  
    | Return number of days in month
- + (int) **weekDay** : (int) year : (int) month : (int) day  
    | Return day of week
- **init** ..... Init empty calendar
- **init** : (int) year ..... Init calendar with year
- **init** : (int) year : (int) month ..... Init calendar with

- | year, month
- (int) **year** ..... Return current year
- **year** : (int) year ..... Set current year
- (int) **month** ..... Return current month
- **month** : (int) month ..... Set current month
- (DData \*) **toData** ..... Get calendar in new data string
- (DText \*) **toText** ..... Get calendar in new text string

#### DConfigReader

##### Methods

- **init** ..... Init config reader
- **free** ..... Free config reader
- (BOOL) **parse** : (id) source : (char \*) name : (id) handler  
    | Parse the source with name, results to handler

#### DConfigWriter

##### Methods

- **init** ..... Init config writer
- **init** : (id) dest ..... Init config writer with config dest.
- **free** ..... Free the config writer
- (BOOL) **startConfig** : (id) dest ..... Start writing config
- (BOOL) **startConfig** ..... Start writing config
- (BOOL) **endConfig** ..... Done writing config
- (BOOL) **section** : (char \*) name ..... Write section in config
- (BOOL) **option** : (char \*) sect : (char \*) opt  
    | : (char \*) val ..... Write option in config
- (BOOL) **comment** : (char \*) comment ..... Write comment
- (void) **error** : (int) number : (char \*) name  
    | : (int) lineNumber : (int) columnNumber .... Report error

#### DConfigTree

##### Methods

- **init** ..... Init an empty config tree
- **init** : (id src : (char \*) name ..... Init config tree with src
- **free** ..... Free config tree
- (BOOL) **read** : (id) srce : (char \*) name  
    | Read config tree from source
- (BOOL) **write** : (id) dest : (char \*) name  
    | Write config tree to dest
- (BOOL) **set** : (char \*) sect : (char \*) opt : (char \*) val  
    | Set/Insert an option for a section
- (char \*) **get** : (char \*) sect : (char \*) opt . Get an option
- (BOOL) **has** : (char \*) sect ..... Check for a section

- (BOOL) **has** : (char \*) sect : (char \*) opt  
    | Check for an option
- (BOOL) **remove** : (char \*) sect ..... Remove a section
- (BOOL) **remove** : (char \*) sect : (char \*) opt  
    | Remove an option
- (DList \*) **sections** ..... Return a list with all sections
- (DList \*) **options** : (char \*) section  
    | Return all options for a section

#### DDiscreteDistribution

##### Methods

- **init** ..... Init discrete distribution
- **free** ..... Free distribution
- (int) **length** ..... Return number values in distribution
- (double) **sum** ..... Return sum values in distribution
- (double) **sumSquared** ..... Return squared sum of values
- (BOOL) **range** : (double) min : (double) max  
    | Add score with range [min,max]
- (BOOL) **range** : (int) value  
    | Add score with range [v-0.5,v+0.5]
- **reset** ..... Reset length, sums and scores
- (BOOL) **update** : (double) value  
    | Update distribution with value
- (double) **mean** ..... Calculate mean of values
- (double) **average** ..... Calculate average of values
- (double) **variance** ..... Calculate variance of values
- (double) **standardDeviation** ..... Calculate SD of values
- (DListIterator \*) **scores** ..... Return iterator on score list

#### DFSM

##### Methods

- **init** ..... Init empty finite state machine
- **copy** ..... Copy FSM
- **shallowFree** ..... Free FSM, excluding states
- **free** ..... Free FSM, including states
- (BOOL) **isChanged** ..... State change after event?
- (DFSMState \*) **current** ..... Get the current state
- (DFSMState \*) **previous** ..... Get the previous state
- **transition** : (DFSMState\*)o:(DBitArray\*)t:(DFSMState\*)d  
    | Add transition from o to d after event in t
- (DFSMState \*) **start** : (DFSMState \*) . Start FSM from state
- (DFSMState \*) **feed** : (int) event ..... Feed event

## DFSMState

### Methods

- **init** ..... Init empty FSM state
- **copy** ..... Copy FSM state
- **free** ..... Free FSM state, including transitions

## DImageHeader

### Constants

- DIM\_UNKN ..... Unknown file format
- DIM\_JPEG ..... JPEG file format
- DIM\_GIF ..... GIF file format
- DIM\_PNG ..... PNG file format
- DIM\_BMP ..... BMP file format
- DIM\_PCX ..... PCX file format
- DIM\_IFF ..... IFF file format
- DIM\_RAS ..... RAS file format
- DIM\_PBM ..... PBM file format
- DIM\_PGM ..... PGM file format
- DIM\_PPM ..... PPM file format
- DIM\_PSD ..... PSD file format
- DIM\_SWF ..... SWF file format

### Methods

- **init** ..... Init an empty image header
- **init** :(id <DDataReadable>) **file** .. Ini with a data readable
- (BOOL) **inspect** :(id <DDataReadable>) **file**  
| Inspect the image via readable
- (char \*) **extension** ..... Return file extension for image
- (char \*) **mime\_type** ..... Return mime type for image
- (DImageHeaderType) **type**... Return the image type DIM...

## DLexer

### Methods

- **init** ..... Init lexer
- **init** :(id) **source** :(char \*) **name** ..... Init lexer with file
- **free** ..... Free the lexer
- (const char \*) **text** ..... Return last scanned text
- (const char \*) **name** ..... Return name current file
- (int) **lineNumber** ..... Return current line nr
- (int) **columnNumber** ..... Return current column nr
- (BOOL) **isEof** ..... Check for end of file reached
- **whiteSpace** :(char \*) **expr** ..... Set white space expr
- **caseSensitive** :(BOOL) **cs** ..... Set case sensitivity

- (BOOL) **source** :(id) **source** :(char \*) **name** .... Start with source
- (BOOL) **popSource** ..... Pop source
- (BOOL) **nextWhiteSpace** ..... Skip white space
- (BOOL) **nextString** :(char \*) **cstring** ..... Scan string
- (BOOL) **nextExpression** :(char \*) **cstring** .... Scan reg exp
- (BOOL) **nextLine** ..... Scan remaining of line
- (BOOL) **checkWhiteSpace** ..... Check white space
- (BOOL) **checkString** :(char \*) **cstring** ..... Check string
- (BOOL) **checkExpression** :(char \*) **cstring** .. Check reg exp
- **next** ..... Move scanner position after check...
- **next** :(unsigned) **positions** ..... Move scanner position
- **error** :(char \*) **msg** ..... Generate error on stderr

## DObjcTokenizer

### Constants.Generaltokens

- DOT\_UNKNOWN ..... Unknown token
- DOT\_EOFF ..... End of file token
- DOT\_EOFL ..... End of line token
- DOT\_WHITESPACE ..... Whitespace token
- DOT\_COMMENT ..... Comment token
- DOT\_IDENTIFIER ..... Identifier token
- DOT\_OPERATOR ..... Operator token
- DOT\_PREPROCESSOR ..... Preprocessor token

### Constants.Literaltokens

- DOT\_DEC\_NUMBER ..... Decimal number token
- DOT\_OCT\_NUMBER ..... Octal number token
- DOT\_HEX\_NUMBER ..... Hex number token
- DOT\_FLP\_NUMBER ..... Floating point number token
- DOT\_CHAR ..... Character token
- DOT\_STRING ..... String token
- DOT\_WIDE\_CHAR ..... Wide character token
- DOT\_WIDE\_STRING ..... Wide string token
- DOT\_OBJC\_STRING ..... Objective-c string token

### Constants.Preprocessortokens

- DOT\_PASSERT ..... Assert token
- DOT\_PDEFINE ..... Define token
- DOT\_PELIF ..... Elseif token
- DOT\_PELSE ..... Else token
- DOT PENDIF ..... Endif token
- DOT\_PERROR ..... Error token
- DOT\_PIDENT ..... Ident token
- DOT\_PIF ..... If token
- DOT\_PIFDEF ..... Ifdef token

- DOT\_PIFNDEF ..... Ifndef token
- DOT\_PINCLUDE ..... Include token
- DOT\_PIMPORT ..... Import token
- DOT\_PLINE ..... Line token
- DOT\_PPPragma ..... Pragma token
- DOT\_PUNASSERT ..... Unassert token
- DOT\_PUNDEF ..... Undefine token
- DOT\_PWARNING ..... Warning token

### Constants.Predefinedtypes

- DOT\_TCHAR ..... Char token
- DOT\_TDOUBLE ..... Double token
- DOT\_TFLOAT ..... Float token
- DOT\_TINT ..... Integer token
- DOT\_TSHORT ..... Short token
- DOT\_TLONG ..... Long token
- DOT\_TUNSIGNED ..... Unsigned token

### Constants.Storageetokens

- DOT\_AUTO ..... Auto token
- DOT\_CONST ..... Const token
- DOT\_EXTERN ..... Extern token
- DOT\_REGISTER ..... Register token
- DOT\_STATIC ..... Static token
- DOT\_VOLATILE ..... Volatile token

### Constants.Typedefinitiontokens

- DOT\_ENUM ..... Enum token
- DOT\_STRUCT ..... Struct token
- DOT\_TYPEDEF ..... Typedef token
- DOT\_UNION ..... Union token

### Constants.Flowkeywords

- DOT\_BREAK ..... Break token
- DOT\_CASE ..... Case token
- DOT\_CONTINUE ..... Continu token
- DOT\_DEFAULT ..... Default token
- DOT\_DO ..... Do token
- DOT\_ELSE ..... Else token
- DOT\_FOR ..... For token
- DOT\_GOTO ..... Goto token
- DOT\_IF ..... If token
- DOT\_RETURN ..... Return token
- DOT\_SWITCH ..... Switch token
- DOT\_WHILE ..... While token

### Constants.Objectiveckeywords

- DOT\_INTERFACE ..... Interface token
- DOT\_IMPLEMENTATION ..... Implementation token
- DOT\_PROTOCOL ..... Protocol token

DOT\_END ..... End token  
DOT\_PRIVATE ..... Private token  
DOT\_PROTECTED ..... Protected token  
DOT\_PUBLIC ..... Public token  
DOT\_SELECTOR ..... Selector token  
DOT\_CLASS ..... Class  
DOT\_ENCODE ..... Encode token  
DOT\_DEFS ..... Defs token  
DOT\_TRY ..... Try token  
DOT\_CATCH ..... Catch token  
DOT\_FINALLY ..... Finally token  
DOT\_THROW ..... Throw token  
DOT\_SYNCHRONIZED ..... Synchronized token  
DOT\_SYNCHRONIZE ..... Synchronize token

*Constants.Logicaloperators*

DOT\_AND ..... And token  
DOT\_OR ..... Or token  
DOT\_NOT ..... Not token

*Constants.Comparisonoperators*

DOT\_SMALLER ..... Smaller token  
DOT\_SMALLER\_EQUAL ..... Smaller equal token  
DOT\_GREATER ..... Greater token  
DOT\_GREATER\_EQUAL ..... Greater equal token  
DOT\_EQUAL ..... Equal token  
DOT\_NOT\_EQUAL ..... Not equal token

*Constants.Bitwiseoperators*

DOT\_BIT\_AND ..... Bit and token  
DOT\_BIT\_OR ..... Bit or token  
DOT\_BIT\_XOR ..... Bit xor token  
DOT\_BIT\_NOT ..... Bit not token  
DOT\_BIT\_LEFT ..... Bit left token  
DOT\_BIT\_RIGHT ..... Bit right token

*Constants.Generaloperators*

DOT\_COLON ..... Colon token  
DOT\_SEMI\_COLON ..... Semi-colon token  
DOT\_COMMA ..... Comma token  
DOT\_BLOCK\_OPEN ..... Block open token  
DOT\_BLOCK\_CLOSE ..... Block close token  
DOT\_BRACE\_OPEN ..... Brace open token  
DOT\_BRACE\_CLOSE ..... Brace close token  
DOT\_BRACKET\_OPEN ..... Bracket open token  
DOT\_BRACKET\_CLOSE ..... Bracket close token  
DOT\_DEREFERNC ..... Dereference token  
DOT\_FIELD ..... Field token  
DOT\_CONDITION ..... Condition token

DOT\_SIZEOF ..... Sizeof token  
DOT\_ADDRESSOF ..... Addressof token  
DOT\_VAR\_ARG ..... Variable arguments token  
DOT\_PCONCAT ..... Concatenation token  
DOT\_PSTRINGIFY ..... Stringify token

*Constants.Assignmentoperators*

DOT\_ASSIGNMENT ..... Assignment token  
DOT\_PLUS\_ASSIGNMENT ..... Plus assignment token  
DOT\_MINUS\_ASSIGNMENT ..... Minus assignment token  
DOT\_MULT\_ASSIGNMENT ..... Multiply assignment token  
DOT\_DIV\_ASSIGNMENT ..... Division assignment token  
DOT\_REM\_ASSIGNMENT ..... Remainder assignment token  
DOT\_LEFT\_ASSIGNMENT ..... Bit left assignment token  
DOT\_RIGHT\_ASSIGNMENT ..... Bit right assignment token  
DOT\_AND\_ASSIGNMENT ..... Bit and assignment token  
DOT\_XOR\_ASSIGNMENT ..... Bit xor assignment token  
DOT\_OR\_ASSIGNMENT ..... Bit or assignment token

*Constants.Arithmeticoperators*

DOT\_INCREMENT ..... Increment operator  
DOT\_DECREMENT ..... Decrement operator  
DOT\_PLUS ..... Plus operator  
DOT\_MINUS ..... Minus operator  
DOT\_STAR ..... Star operator  
DOT\_VALUEOF ..... Value of operator  
DOT\_DIVISION ..... Division operator  
DOT\_REMAINDER ..... Remainder operator

*Classmethods*

+ (BOOL) isLogicalOperator :(int) token  
| Check for logical operator token  
+ (BOOL) isComparisonOperator :(int) token  
| Check for comparison operator token  
+ (BOOL) isBitOperator :(int) token  
| Check for bit operator token  
+ (BOOL) isAssignmentOperator :(int) token  
| Check for assignment operator token  
+ (BOOL) isArithmeticOperator :(int) token  
| Check for arithmetic operator token  
+ (BOOL) isOperator :(int) token  
Check for an operator token  
+ (BOOL) isPredefinedTypeKeyword :(int) token  
| Check for a predefined type token  
+ (BOOL) isStorageKeyword :(int) token  
| Check for a storage keyword token  
+ (BOOL) isTypeDefinitionKeyword :(int) token  
| Check for type definition token  
+ (BOOL) isFlowKeyword :(int) token

| Check for a flow keyword token  
+ (BOOL) isKeyword :(int) token  
.. Check for a keyword token  
+ (BOOL) isObjcKeyword :(int) token  
| Check for an objc-keyword  
+ (BOOL) isDirective :(int) token  
.. Check for directive token  
+ (char \*) operator :(int) token  
Return operator description  
+ (char \*) keyword :(int) token  
.. Return keyword description  
+ (char \*) objcKeyword :(int) token  
| Return objc-keyword description  
+ (char \*) directive :(int) token  
| Return directive description

*Objectmethods*

- (DObjcTokenizer \*) init ..... Init default tokenizer  
- init :(id) source :(char \*) name  
Init tokenizer with source  
- free ..... Free the tokenizer  
- (char \*) text ..... Return the last scanned text  
- (char \*) name ..... Return the name of the source  
- (int) lineNumber ..... Return the current line number  
- (int) columnNumber ..... Return the current column number  
- (BOOL) isEof ..... Check if the end of the source is reached  
- (BOOL) skipWhiteSpace  
Check if white space must be skipped  
- skipWhiteSpace :(BOOL) skip ..... Set skip white space  
- (BOOL) source :(id) source :(char \*) name  
| Start tokenizing source  
- (BOOL) popSource ..... Pop source for sources stack  
- (int) nextToken ..... Scan source for next token  
- (int) checkToken ..... Check source for next token  
- next ..... Move source location after checkToken  
- error :(char \*) msg ..... Generate error on stderr

DProperty

*Methods*

- init ..... Init empty property  
- init :(char \*) name :(id) value ..... Init property  
- init :(char \*) name ..... Init property group  
- deepen ..... Deepen a property after copy  
- free ..... Free a property  
- (BOOL) isGroup ..... Check if property is a group  
- (const char \*) name ..... Return name of property  
- (id) value ..... Return value reference of the property  
- property :(char \*) name :(id) value ..... Set the property  
- group :(char \*) name ..... Set the property group

## DPropertyTree

### Methods

- **init** ..... Init default property tree
- **init** :(char \*) **name** ..... Init named property tree
- **free** ..... Free the property tree
- (char \*) **name** ..... Return the name of the tree
- **name** :(char \*) **name** ..... Set the name of the tree
- **group** :(DProperty \*) **parent** :(char \*) **name**
  - | Add group property to the property tree
- **property** :(DProperty \*) **parent** :(char \*) **name**
  - |:(id) **value** ..... Add property to the property tree
- (BOOL) **read** :(id) **source** :(char \*) **name**
  - | Read the property tree and set the value references
- (BOOL) **write** :(id) **destination** :(char \*) **name**
  - | Write the property tree by reading the value references
- (BOOL) **remove** :(DProperty \*) **property** .. Remove property
- (BOOL) **remove** :(DProperty \*) **parent** :(char \*) **name**
  - | Remove property by name

## DSource

### Methods

- **init** ..... Init empty source
- **init** :(id) **source** :(char \*) **name** ..... Init source with file
- **free** ..... Free the source
- (const char \*) **name** ..... Return the name current source
- (const char \*) **line** ..... Return remaining line
- (int) **lineNumber** ..... Return current line number
- (int) **columnNumber** ..... Return the current column number
- (BOOL) **set** :(id) **source** :(char \*) **name** . Start with source
- (BOOL) **nextLine** ..... Get the next line
- (BOOL) **appendLine** ..... Append the next line
- (BOOL) **scanned** :(unsigned) **lengthSet** length chars scanned
- (BOOL) **isLineScanned** ..... Check if full line scanned
- (DSource \*) **error** :(char \*) **msg** .. Generate error on stderr

## DTokenizer

### Methods

- **init** ..... Init tokenizer
- **init** :(id) **source** :(char \*) **name** ... Init tokenizer with file
- **free** ..... Free the tokenizer
- (char \*) **text** ..... Return last scanned text
- (const char \*) **name** ..... Return name current file
- (int) **lineNumber** ..... Return current line number

- (int) **columnNumber** ..... Return current column number
- (BOOL) **isEof** ..... Check for end of file
- **skipWhiteSpace** :(BOOL) **skip** ..... Set skip white space
- (BOOL) **skipWhiteSpace** ..... Return skip white space
- (BOOL) **source** :(id) **source** :(char \*) **name** .... Start with source
- (BOOL) **popSource** ..... Pop source from sources stack
- (int) **nextToken** ..... Scan source for token
- (int) **checkToken** ..... Check source for token
- **next** ..... Move source location after checkToken
- **next** :(unsigned) **positions** ..... Move source location
- **error** :(char \*) **msg** ..... Generate error on stderr

### Basicscanners to be overridden

- (DText \*) **whiteSpace** :(char \*) **text** White space scanner
- (DText \*) **comment** :(char \*) **text** ..... Comment scanner
- (DText \*) **keyword** :(char \*) **text** ..... Keyword scanner
- (DText \*) **string** :(char \*) **text** ..... String scanner
- (DText \*) **operator** :(char \*) **text** ..... Operator scanner
- (DText \*) **number** :(char \*) **text** ..... Number scanner
- (DText \*) **extraToken1** :(char \*) **text** ..... Extra scanner
- (DText \*) **extraToken2** :(char \*) **text** ..... Extra scanner
- (DText \*) **extraToken3** :(char \*) **text** ..... Extra scanner

## DXMLElement

### Constants

- DXML\_ELEMENT ..... XML Element
- DXML\_ATTRIBUTE ..... XML Attribute
- DXML\_TEXT ..... XML Text
- DXML\_CDATA ..... XML CDATA Text
- DXML\_PI ..... XML Processing Instruction
- DXML\_COMMENT ..... XML Comment
- DXML\_DOCUMENT ..... XML Document
- DXML\_NAMESPACE ..... XML Namespace

### Methods

- **init** ..... Init empty xml node
- **init** :(int) **type** :(char \*) **name** :(char \*) **value**
  - | Init xml node with type, name and value
- **deepen** ..... Deepen the copied node
- **free** ..... Free the node
- (int) **type** ..... Return the type of the node
- (char \*) **name** ..... Return the name of the node
- (char \*) **value** ..... Return the value of the node
- **set** :(int) **type** :(char \*) **name** :char \*) **value**
  - | Set the type, name and value of the node

## DXMLElement : DTree

### Methods

- (DXMLElement \*) **init** ..... Init empty xml tree
- **init** :(id) **source** :(char \*) **name** :(char) **separator**
  - | Init xml tree with xml source
- **free** ..... Free the tree and the stored nodes
- **shallowFree** ..... Free the tree, not the stored nodes
- (BOOL) **read** :(id) **source** :(char \*) **name** :(char) **sep**
  - | Build xml tree from xml source
- (BOOL) **write** :(id) **destination** :(char \*) **name**
  - | Write xml tree to destination

## DXMLElementWriter

### Methods

- **init** ..... Init default xml writer
- **init** :(id) **destination** :(char) **separator**
  - | Init xml writer with a destination
- **free** ..... Free the writer
- (BOOL) **start** :(id) **destination** :(char) **separator**
  - | Start the writing of a xml file
- (BOOL) **startDocument** :(char \*) **version** :(char \*) **enc**
  - |:(int) **standalone** ..... Write start of document
- (BOOL) **endDocument** ..... Write end of document
- (BOOL) **startElement** :(char \*) **name** . Write start of element
- (BOOL) **attribute** :(char \*) **attr** :(char \*) **value**
  - | Write attribute
- (BOOL) **endElement** ..... Write end of element
- (BOOL) **characters** :(char \*) **text** ..... Write text
- (BOOL) **comment** :(char \*) **text** ..... Write comment
- (BOOL) **processingInstruction** :(char \*) **target**
  - |:(char \*) **value** ..... Write PI
- (BOOL) **startCDATA** ..... Write start of CDATA section
- (BOOL) **endCDATA** ..... Write end of CDATA section
- (BOOL) **startNamespace** :(char \*) **prefix** :(char \*) **uri**
  - | Write start of namespace
- (BOOL) **endNamespace** ..... Write end of namespace
- (BOOL) **unparsed** :(char \*) **text** ..... Write unparsed text
- (void) **error** :(int) **number** :(const char \*) **name**
  - |:(int) **lineNumber** :(int) **columnNumber** .... Report error

Version 0.7.0. This card may be freely distributed under the terms of the GNU general public licence

Copyright © 2003-2005 by Dick van Oudheusden