Context: **ILNumerics.ILMath**

Array Creation I

A = ...

one value for all elements
array<T>(T, params int[])
array<T>(T, ILSize)

initialize all values
array<T>(params T[])
array<T>(T[], ILSize)
array<T>(T[], params int[])

initialize all default(T)
zeros<T>(params int[])
zeros<T>(ILSize)
zeros(NumericType,ILSize)

initialize all 0.0 (double)
zeros(params int[])
zeros(ILSize)

initialize all one
ones(NumericType,ILSize)
ones<T>(params int[])
ones<T>(ILSize)

initialize all 1.0 (double)
ones(params int[])
ones(ILSize)

Array Creation II

empty<>	(sized) empty array
rand randn	random numbers normal distribution
eye<> diag<> repmat	identity matrix diagonal matrix replicate array
linspace<> logspace<> arange<>	linear spaced vector logar. spaced vector regul. spaced vector

Array Properties

C	Fast, lazy clone
S	Size descriptor
T	Transpose
Length	Longest dimension length

IsEmtpy, IsScalar, IsMatrix
IsVector, IsColumnVector
IsRowVector

IsComplex, IsNumeric

Array Methods

Clone()	lazy clone
Concat(..)	concatenation
Reshape(..)	reshaped
Shifted(..)	shifted
Subarray(..)	get subarray
Equals(..)	recursiv eq
ExportValues()	values ➡ T[]
GetLimits(..)	min/max val
GetValue(..)	get single val
SetValue(..)	set single val
SetRange(..)	alter a range

Subarray Creation/ -Assignment B[...] = A[...]

ILNumerics uses zero based indexing !!

Ways to address a single dimension:

	By strings	Fast (w/o string)
full dimension	“:“	full
single index	“4“	4
mult. indices	“2,3,4“	cellv(2,3,4)
Range, step:1/3	“2:4“ “2:3:4“	r(2,4), r(2,3,4)
end	“end“	end
rel. to end		end / 2 - 1
use variable i, A		i, cellv(2,3,i), r(0,i), r(i,end), A, A[...],…
combined (yet addressing one dimension)	“:,4,2:4,end“	cellv(full,4,r(2,4), end)
dimension seperator	; “4;2:4“ but: “4“,“2:4“	,
mixed mode	B[“2,3,4“,r(i,end/2),“:,0“]	

Array Types

	Dense Array	Cell Array	Logical Array	Re-/Assign
Local Arrays A	Array<T>	Cell	Logical	A = … ;
Input Parameter inA[1]	InArray<T>	InCell	InLogical	inA = …; [1]

Out Parameter outA	OutArray<T>	OutCell	OutLogical	outA.a = …;
				outA.Assign(..)

Return Values	RetArray<T>	RetCell	RetLogical	

[1] *immutable array, only new array assignable to variable*