CASE – SYMBOLS

The Mobile Compiler

This document is meant to document all the language symbols which is necessary for preprocessing

Symbol List:

\_

->

~>

.

…

~

,

;

:

::

??

.

“

\_

$

(

( c )

( m )

()  
endclass  
.get

.set

.getclass

.forName

get

set

?operator (+operator or –operator or moperator)

)

[

[]

[][]

]

{

}

#

\*

\*/

/\* \*/

/\*\*

//

\

\?  
<<

\’

\”

\\

\|

\A

\a  
\b  
\f  
\n  
\r  
\t  
\v \nnn \xnn...

\b

\f

\n

\r

\t

\t

\v

&

&

&=

#

%%

%

^

+

<

<?>

<=

<>

<T>  
caller

=

==

=$

>

>>

<>

<[ ]>

<[ ][ ]>

[ ]

[ ] [ ]

<=

>=

Div

Mod

Rem

(<<)

(>>)

++

--

+=

-=

/=

\*=

`

‘

.

{ }

<=

=>

,

||

endclass

exp

abs

matrix

vector

convo

car

cdr

|

|

||

~

( c )

@:injected

@application

@selfmodifier

@texts:

@module

@controller

@table

abstract

also

and

array

ArrayList

Arrays

as

assert

auto

autofall

autorelease

assigner

await

atom

base

Boolean

bind

bowtie

branch

break

break

bytecode

callbybreaction

car

case

catch

cdr

caller

Char

class

Collection

commandExecute

cons

const

const

construct

continue

controller

Database

d

debug

Decimal

deep

default

del

diagram

do

Double

DoublyLinkedArrayList

DoublyLinkedList

else

enum

EntityPool

equiv

exit

explicit

extend

extern

f

fall

false

false

final

finally

filter

Fixed

Float

flux

for

foreach

friend

frozen

function

future

fx

get

getLine

getChar

getString

getType

generator

global

goto

goto

HashTables

i

if

implements

implicit

import

in

include

inclusion

inherit

inline

instanceOf

Integer

Iterator

interface

internal

injector

is

isDigit

isfalse

isLetter

isLowerCase

isnot

isnothing

istrue

isUpperCase

isWhiteSpace

Iterator

jump

l

label

lambda

last

len (length of a string)

leap

like

Linked List

Lists

lock

Long

Maps

meta

model

MultiMaps

Namespace

native

need

new

not

notand

notor

null

Objects

operator

or

onewaybind

package

pass

pass

pack

printbug

private

procedure

protected

public

publisher

promise

Queue

range

register

release

rename

residue

reside

resolve

ret

retain

refactor

release

return

requires

router

sealed

static

stateful

satisfies

set

Sets

shall

Short

sizeof

Sort Library

Stacks

Stateful

singleton

stream

Strings

struct

subscriber

super

success

switch

synchronized

$scope

syn

tag

tail

target

ties

texts:

thaw

this

throw

throws

tie

toLowerCase

toString

toUpperCase

Trees

true

true

try

typedef

typeOf

Uint

Ulong

until

Ushort

union

Vector

virtual

view

void

void

volatile

wait

Write

WriteLn

wait

where

while

xor