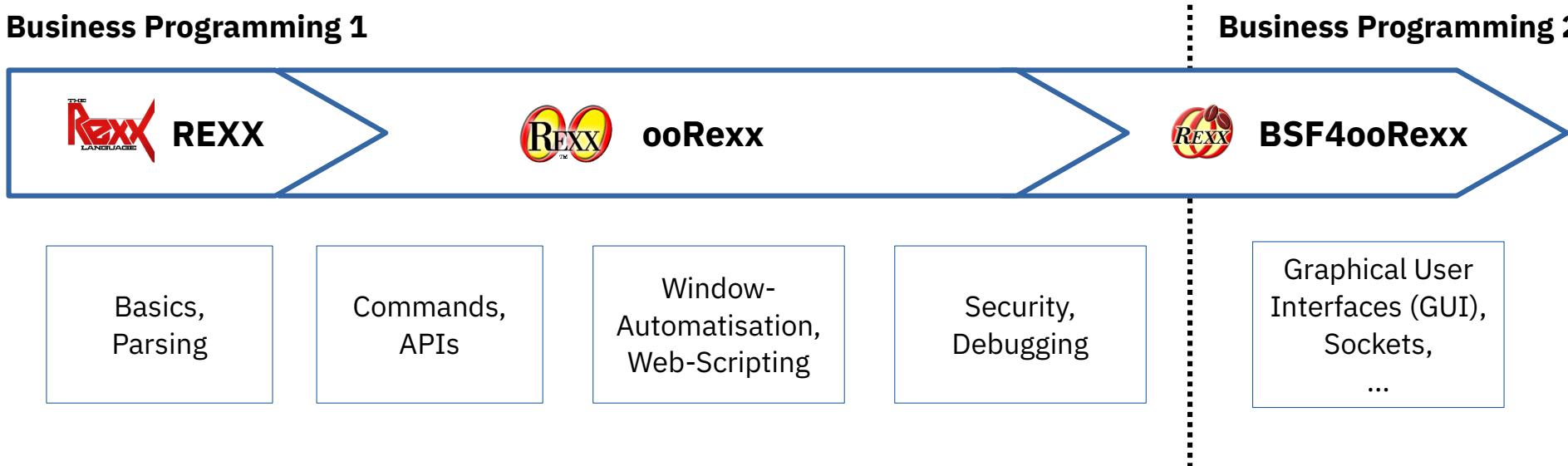


Procedural and Object-oriented Programming 6

ooRexx Runtime Environment and Resolving Environment Symbols

Business Programming 1



Environment Symbols

- Definition of an environment symbol
 - Starts with a dot
 - Followed by a RerrMsg symbol
 - Some examples for environment symbols with their resolved values:

```
say .true      -- yields: 1 (environment constant)
say .false     -- yields: 0 (environment constant)
say .nil       -- yields: The NIL object (environment constant)
say .he.ho     -- yields: .HE.HO (not found, environment symbol in uppercase)
say .bag        -- yields: The Bag class (from REXX package, also .environment)
say .output    -- yields: The OUTPUT monitor (from .local)
.context~package~local~my.test="This is a test" -- add an entry to package's local
say .my.test   -- yields: This is a test
```
- How does ooREXX arrive at these values?

Resolving Environment Symbols, 1



1. First resolving the constants

- **.true** (1), **.false** (0) or **.nil** (*The NIL object*)

2. Next removing the leading dot, remaining symbol used for lookup

1) Resolving classes

- (1) Classes defined in the current program/package
- (2) Public Classes of required programs
- (3) ooRexx classes (REXX package) which also get stored in the **.environment** directory

2) Resolving entries by looking up the following environment directories one by one

- (1) The **local** package directory of the current program (i.e.: **.context~package~local**)
- (2) The ooRexx **.local** environment directory (each ooRexx interpreter has a different one)
- (3) The ooRexx global **.environment** directory (all ooRexx interpreters share this one)
- (4) The ooRexx (unnamed) system directory

Resolving Environment Symbols, 2



3. Next resolving the ooRexx runtime environment symbols

- **.CONTEXT, .LINE, .METHODS, .RESOURCES, .ROUTINES, or .RS**

4. Finally, if not found returning the environment symbol in uppercase

→ Note: ooRexx reserves symbols without a dot for itself

- Hence, if you store entries in the package **local**, **.local** or global **.environment** directories then make sure that the *index value* (name without leading dot) you use contains at least one dot!

Example:

```
.local~my.secret="None of your business! ;)" -- store in .local directory
say ".my.secret:" .my.secret -- displays: .my.secret: None of your business! ;)
```