

# EXTENDING THE FRAMEWORK

Tobias Bussmann  
MBS Xojo Developer Conference  
Koblenz, 19.09.2014

# WHAT'S THIS ABOUT?

# PLUGINS?

MBS, Einhugur, Jérémie, TT, ...

[http://docs.xojo.com/index.php/Third\\_Party\\_Products](http://docs.xojo.com/index.php/Third_Party_Products)

# DECLARES?

MacOSLib, Windows Functionality Suite, ...

[http://docs.xojo.com/index.php/Community\\_Resources](http://docs.xojo.com/index.php/Community_Resources)

# WHAT'S THIS ABOUT?

Using the built in ways to enhance  
the framework according to your needs



# TOPICS

- The Extends-Keyword
- Control subclasses
- Subclassing
- Wrapper Classes
- Documentation
- Code Management

# BENEFITS

- Ease coding / convenience / readability
- Avoid to do things over and over again
- Have a central place for your helper functions

# EXTENDS-KEYWORD

- add your code to foreign class
- no overriding, public scope
- not possible to extend a shared method, operator\_ methos
- What you can extend:
  - (Framework) classes
  - Controls
  - intrinsic Types
  - Arrays
  - even Interfaces



# DEMO

- Extend String with „Contains“
- Use ParamArray for „InList“
- direct Accessor for Objects (and Arrays!) in Variants
- Integer Array with „Multiply-by“
- Get Dictionary Value by Index
- TextArea with AppendLine Method
- Retrieve the Class name as String

# CONTROL SUBCLASSES

- change behavior overall  
e.g. add logging, defaults, fix bugs, enhance
- intercept Event handling:
  - redefine and reraise Event
- Inspector Behavior
- shadow properties with computed properties:
  - cast to superclass instead of call super.

# DEMO

- Cocoa Popup menu with Styles
- WebFileUploader with file type limitation
- WebButton with Action logging  
and different caption when disabled

# SUBCLASSING

- Requires different instantiation
- change behavior of existing Class Methods
- Add Properties
- Operator\_Subscript / \_Lookup:
  - Array-Like or Object-Like interface
  - assigns Keyword



# DEMO

- Database Class that raises Exceptions on error to avoid checking of Error property after each statement



# WRAPPER CLASS

- Especially helpful if you don't have the control over instantiation
- Holds an object of its super class, mimic their interface
- with `Operator_Convert`:
  - convert back and forth to it's base class  
(fake up *and down* casting)
- Can be subclass as well
  - allows `isA` operator, real upcasting (overriding)
  - copy the super class object in `Operator_Convert` instead of wrapping it

# DEMO

- TextOutputStream which handles Encoding like TextInputStream does
- Dictionary or Recordset, that can be accessed like an array:  
Dict(„Key“) / RS.Field

# DOCUMENTATION

- important, esp. if working in team
- common structure, naming helps
- code completion wonky, help/goto-links

# CODE MANAGEMENT

- levels of reusability
- organization in modules per extended thing
- extend where possible, subclass where necessary
- use compatibility flags
- SVN external
- test, update and keep current



Thank you