

WireBox Standalone



By George Murphy

About Me



DeSign By George, Inc.

- Contractor
- ColdFusion since version 4.5
- Member Team ColdBox
- ColdFusion, JavaScript, and Flex developer



WireBox Dependency Injection

1

What is WireBox?

2

How can WireBox help me?

3

WireBox system Requirements?

4

What are the WireBox Flavors?

5

Install and use WireBox?

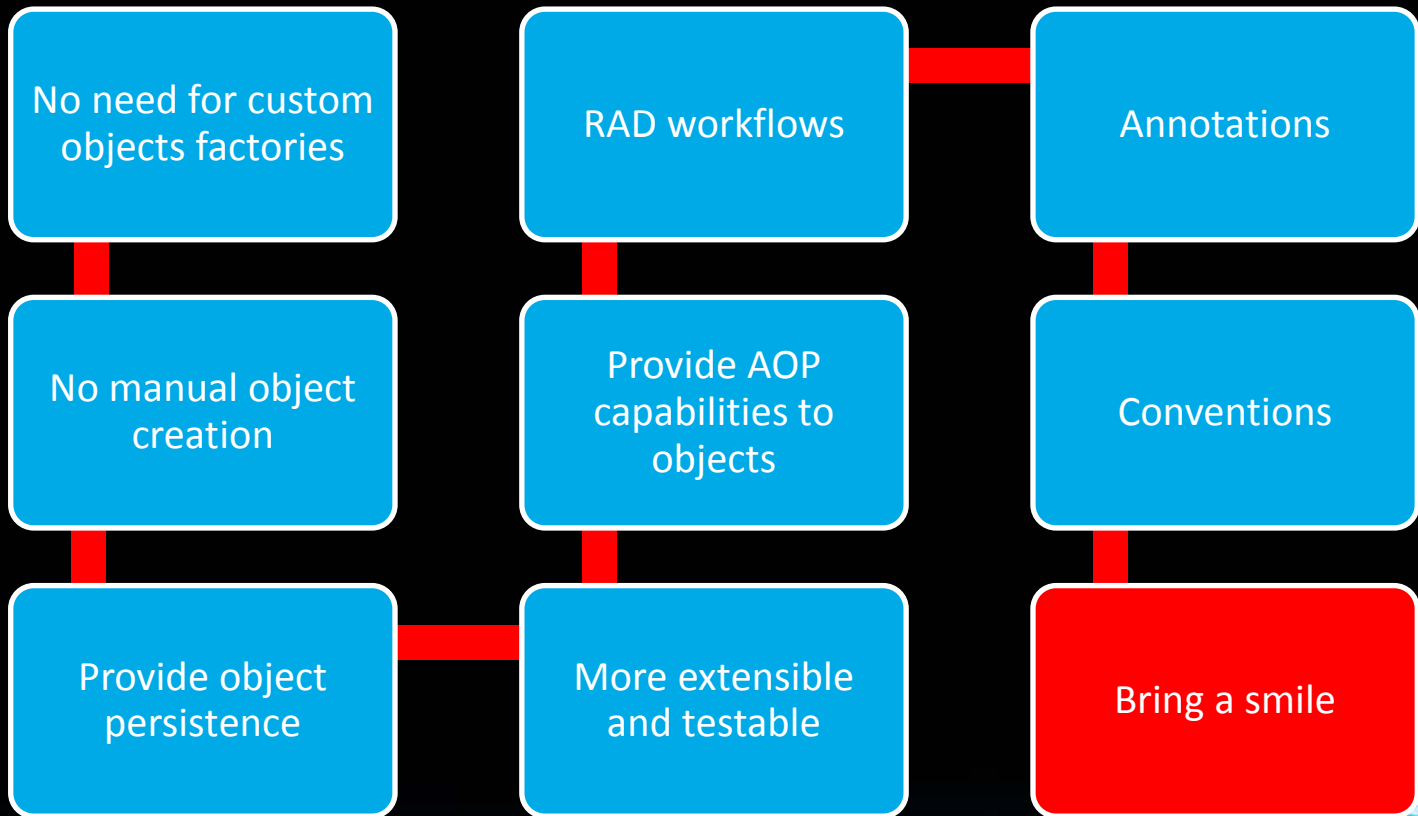


What is WireBox?

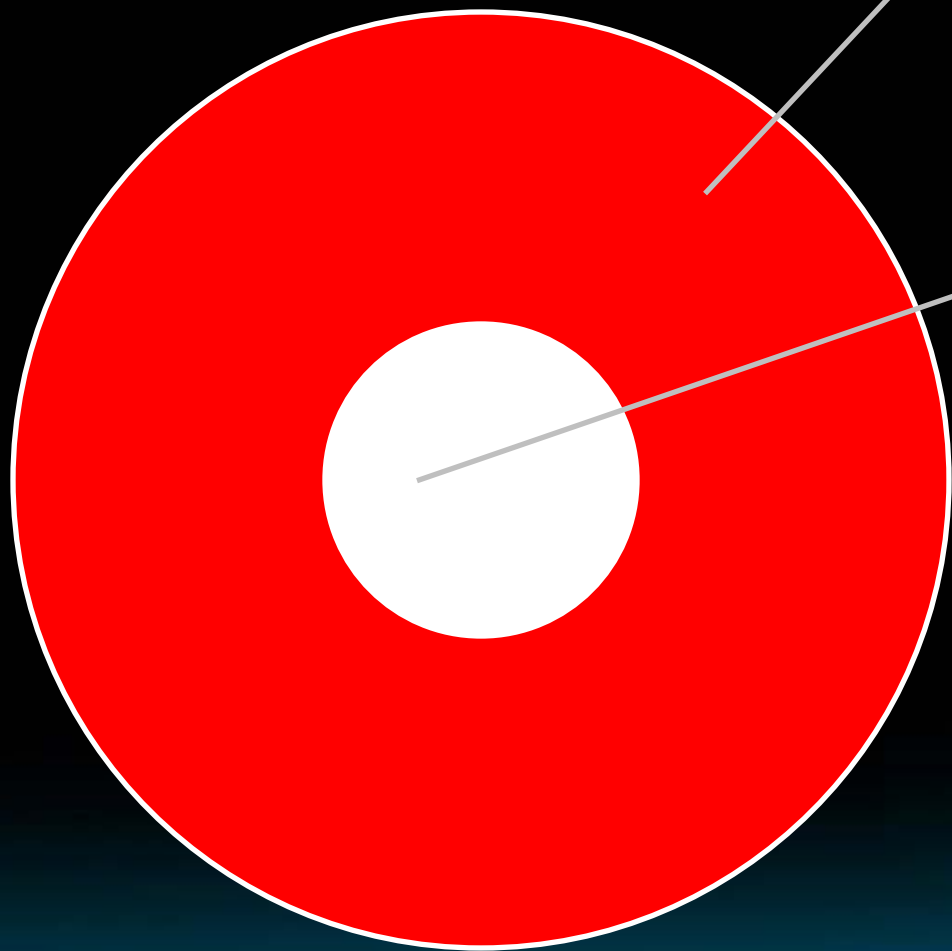
WireBox is a next generation conventions based dependency injection and AOP (Aspect Oriented Programming) framework for ColdFusion.



How can WireBox help me?



WireBox Flavors

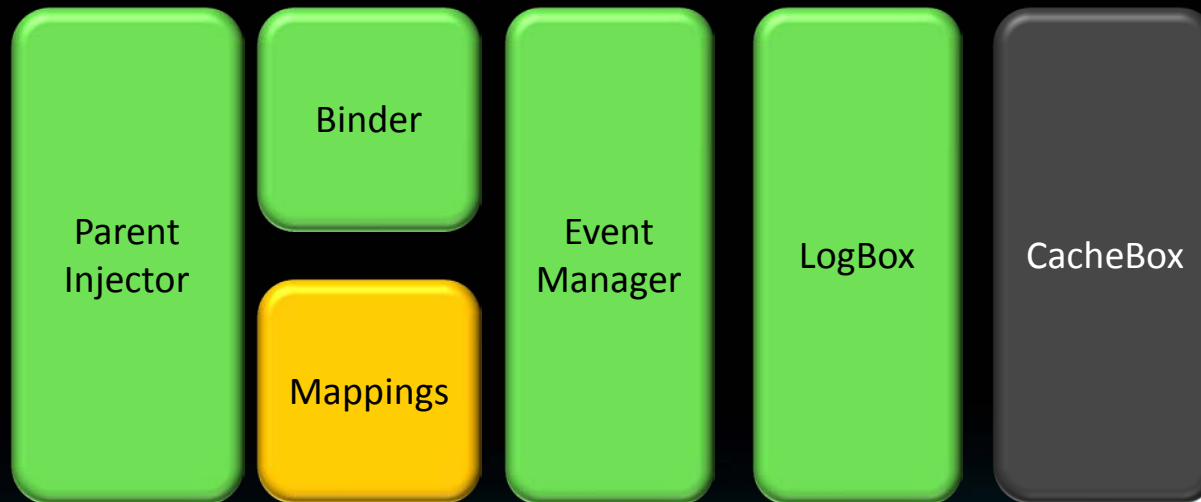


Cherry
ColdBox

Vanilla
Standalone



WireBox Standalone



System Requirements

Download <http://www.coldbox.org/download>



*ColdFusion 8
and above*

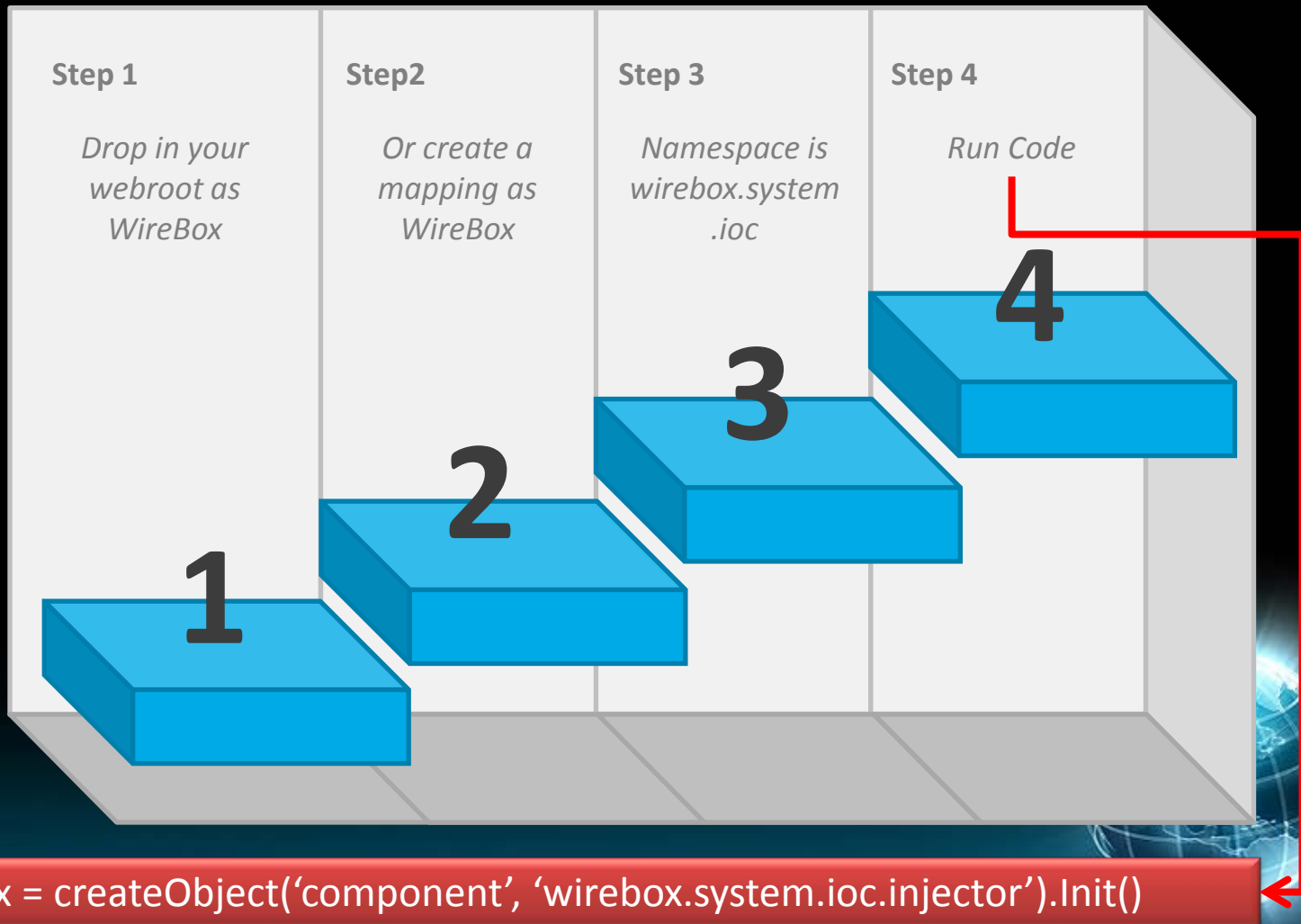
*Railo 3.1 and
above*

*Application
Scope*

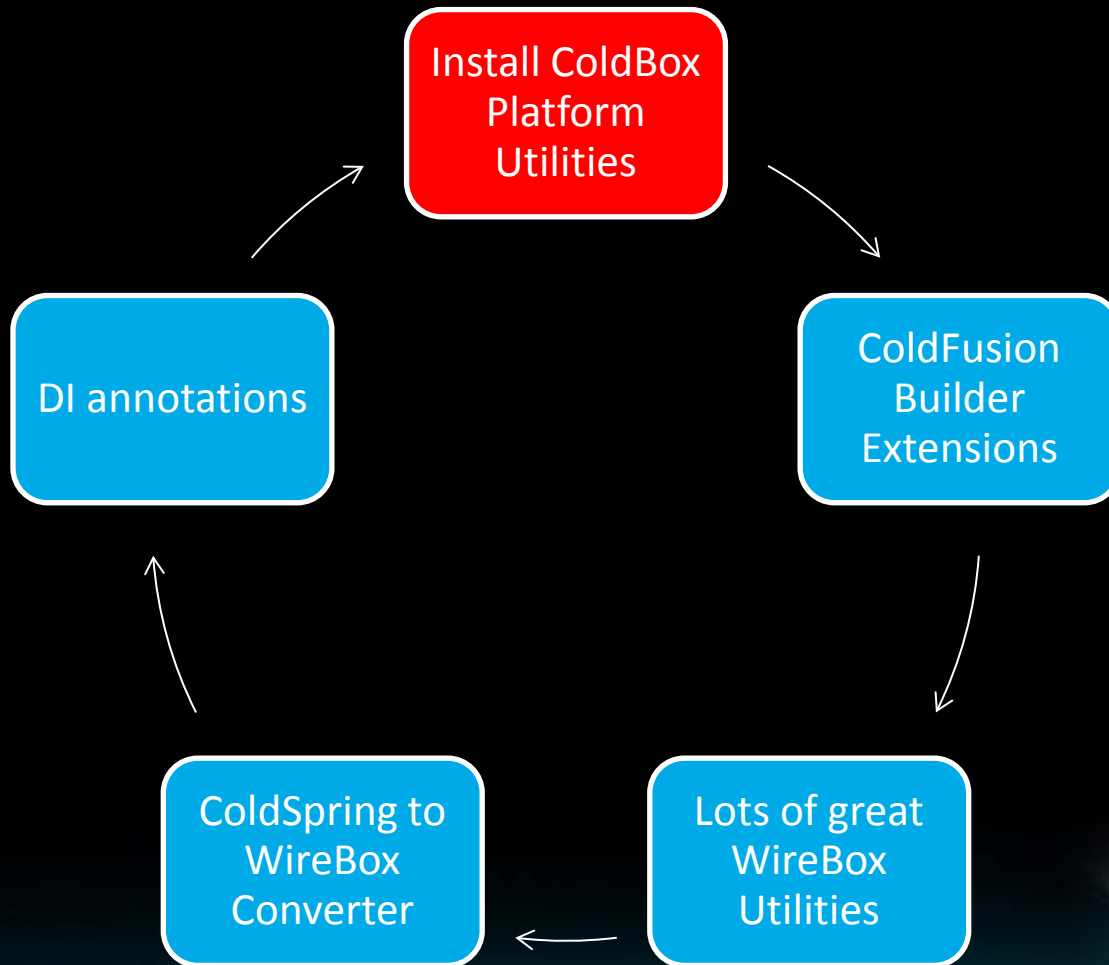
*User can
choose other
scopes*



Install and use WireBox



Make your life easier



<http://www.coldbox.org/forgebox/view/coldbox-platform-utilities>



WireBox Injection Idioms

Constructor argument

```
Component {  
    function init (mydependency) {  
        variables.mydependency = arguments.mydependency;  
    }  
}
```

Setter/Method

```
function setMyDependency (mydependency) {  
    variables.mydependency = arguments.mydependency;  
}
```

Mixins (CFProperty)

```
Component {  
    property name="mydependency" inject;  
}
```



WireBox Creation Styles

Approach	Purpose	Positives	Negatives
Implicit Mappings	To replace Create object Calls	Very fast to prototype Request object by path	Not refactoring friendly



WireBox Creation Styles

Implicit Creation

```
wirebox = createObject("component", "wirebox.system.ioc.injector").init();
```

```
cars = wirebox.getInstance("model.myCarObj").getCars();
```



WireBox Creation Styles

Approach	Purpose	Positives	Negatives
Implicit Mappings	To replace Create object Calls	Very fast to prototype Request object by path	Not refactoring friendly
Explicit Mappings	To replace Create object Calls with named keys	Ability to create multiple named mappings that point to the same class refactoring friendly	Not as fast, must be defined in the binder



WireBox Creation Styles

Explicit Mappings

Explicit Binder Configuration

```
map("koolCars").to("model.myCarObj");
```

```
wirebox = createObject("component", "wirebox.system.ioc.injector").init();
```

```
cars = wirebox.getInstance("koolCars").getCars();
```



WireBox Creation Styles

Approach	Purpose	Positives	Negatives
Implicit Mappings	To replace Create object Calls	Very fast to prototype Request object by path	Not refactoring friendly
Explicit Mappings	To replace Create object Calls with named keys	Ability to create multiple named mappings that point to the same class refactoring friendly	Not as fast, must be defined in the binder
Scan Locations	CFC discovery by conventions	A partial instantiation path is mapped, very fast to prototype	More difficult concept



WireBox Creation Styles

Scan Locations

Scan Locations binder configuration

```
wirebox.scanlocations = ["model"];
```

Scan Locations Creation

```
wirebox = createObject("component", "wirebox.system.ioc.injector").init();
```

```
cars = wirebox.getInstance("myCarObj").getCars();
```



WireBox more than just CFCs

Type	Description
CFC	ColdFusion Components
Java	Any Java Object
Webservice	Any WSDL
RSS	Any RSS or Atom feed
DSL	Any registered or core DSL
Constant	A constant value
Factory	An Object from a factory
Provider	A registered provider object

Let's look at code



WIREFBOX

By ColdBox



Resources

- Wirebox download
<http://www.coldbox.org/download>
- Documentation
<http://wiki.coldbox.org/wiki/WireBox.cfm>
- Sample Code
<https://github.com/murpg/wireboxstandaloneaop>



**Do you have
any questions?**



Contact me



George Murphy

georgemurphy@websbygeorge.com

