



**HAL**  
open science

## **Tornado: A Run-Fail-Grow approach for Dynamic Application Tayloring**

Guillermo Polito, Stéphane Ducasse, Noury Bouraqadi, Luc Fabresse

► **To cite this version:**

Guillermo Polito, Stéphane Ducasse, Noury Bouraqadi, Luc Fabresse. Tornado: A Run-Fail-Grow approach for Dynamic Application Tayloring. 2014. hal-00996908v1

**HAL Id: hal-00996908**

**<https://hal.science/hal-00996908v1>**

Submitted on 27 May 2014 (v1), last revised 15 Jul 2014 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MINES-TELECOM INSTITUTE, MINES DOUAI  
RMOD INRIA LILLE NORD EUROPE

TECHNICAL REPORT

---

**Tornado:**  
**A Run-Fail-Grow approach**  
**for**  
**Dynamic Application Tailoring**

---

*Author:*  
Guillermo POLITO  
*Version:* 1.0

*Supervisors:*  
Stéphane DUCASSE  
Noury BOURAQADI  
Luc FABRESE

May 22, 2014

## Abstract

Producing a small deployment version of an application is a challenge because static abstractions such as packages cannot anticipate the use of their parts. As such, an application often occupies more memory than actually needed. To solve this problem we propose Tornado, a technique to dynamically tailor applications to only embed code (classes and methods) they use. Tornado uses a run-fail-grow approach to prepare an application for deployment. It launches minimal version of an application and installs a minimal set of statements that will start the user's application. This application is run and these statements are executed. When the application fails because there are classes or methods missing, the necessary code is installed. The application is executed until it reaches a stable point, allowing possibly human interaction for applications with UIs. Thus, Tornado creates minimal memory footprint versions of applications by tailoring the whole application's code, including run-time and third party libraries.

We used Tornado to tailor two different applications. We succeeded to tailor a *hello world* application to occupy 1% of its original size. We also experimented with a Seaside web application tailoring in one case only the application's and framework's code and the whole application's code in the other case. In this latter example, we reached memory savings of about 97%. In this report we present an overview on Tornado, and we give details of the results we obtained.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Unused Code Units by Example . . . . .	3
1.2	Challenges of Application Tailoring . . . . .	4
1.3	Tornado in a Nutshell. . . . .	6
<b>2</b>	<b>The Run-Fail-Grow Approach</b>	<b>6</b>
2.1	Run-Fail-Grow with an example . . . . .	8
2.2	Tornado’s architecture . . . . .	10
2.3	Detecting Missing Code Units with a Run-Fail approach . . . . .	11
<b>3</b>	<b>Chosen Technology: Pharo</b>	<b>12</b>
<b>4</b>	<b>The object runtime manipulation interface: Oz object spaces</b>	<b>13</b>
<b>5</b>	<b>The Execution Traps: Advanced Intercession with Ghost-like Proxies</b>	<b>13</b>
<b>6</b>	<b>Object Installation and Object Mappings</b>	<b>14</b>
<b>7</b>	<b>Handling Tailoring Levels through Seeds</b>	<b>15</b>
7.1	Loading an already existing memory snapshot . . . . .	15
7.2	Creating all seed code units from scratch . . . . .	15
<b>8</b>	<b>Preparing the Application for Deployment</b>	<b>16</b>
8.1	Snapshot . . . . .	16
8.2	Building a Static Description . . . . .	16
<b>9</b>	<b>Results</b>	<b>16</b>
9.1	Methodology . . . . .	16
9.2	Hello World Application . . . . .	17
9.3	Seaside Web Application . . . . .	17
<b>10</b>	<b>Conclusion</b>	<b>18</b>
<b>A</b>	<b>Appendix: Method List of a Nurtured Hello World Application</b>	<b>21</b>
<b>B</b>	<b>Appendix: Entry Points to Tailor the Seaside Web Application</b>	<b>23</b>
<b>C</b>	<b>Appendix: Method List of Seaside Counter Application with Full Pharo Seed</b>	<b>24</b>
<b>D</b>	<b>Appendix: Method List of Seaside Counter Application with Empty Seed</b>	<b>30</b>

# 1 Introduction

Deployed object-oriented applications often contain *code units* (e.g. packages, classes, methods) that the running application never uses. This problem shows itself more evident and harder to control under the usage of third party software. Third party libraries and frameworks are designed in a generic fashion that allows multiple usages and functionalities, while applications use only few of them. Examples are logging libraries, web application frameworks or object-relational mappers.

Unused deployed code units have an undesired impact when targeting a constrained infrastructure. Constrained devices may present restrictive hardware such as low primary or secondary memory, or even software impositions such as the Android's Dalvik VM restriction to deploy only 65536 methods<sup>1</sup>. Big JavaScript mashup applications have an impact on loading time due to network speed and parsing time. These limitations may forbid the deployment of applications that contain lots of code units, or limit the amount of applications and content an user can have in its device.

Existant solutions to this problem propose the extraction of used code units of an application to reduce their size and memory footprint. Java Micro Edition [8] proposes a general purpose specialized runtime environment with no possibility of customization. Other solutions in the field propose to automatically detect and extract used code units, so called *tailoring*, with static call graph construction as the most dominant technique [7]. Static approaches present limitations in the presence of dynamic features such as reflection or in the absence of static type annotations. Additionally, they are in general designed to extract all used code units with no possibility for the user to customize the process of selection.

## 1.1 Unused Code Units by Example

To clearly show the problem, consider the application using a logging library in Figure 1. An interface is present in the diagram to show polymorphism between two classes that do not share a class inheritance hierarchy. However, some languages, such as the dynamically typed ones, may not need to represent it in the source code.

Figure 2 shows the code of this application, written in the Pharo Smalltalk language. This application contains a `MainApp` class with a `start` method, which is the entry point of our application. The `start` method creates an instance of `StdoutLogger` and logs the application's start and end. In turn, the `StdoutLogger` uses the `stdout` global instance to log in the standard output the current time and the message. To print the time, the `StdoutLogger` makes use of the `Time` class from the base libraries of the language. Note that for the sake of clarity, we didn't include in the example all base libraries, though, in modern programming languages they represent a large codebase with several features going from networking to multithreading. For example, Java 8 SE contains 4240 classes<sup>2</sup>, and the development edition of Pharo 2.0 contains 3342 classes and traits.

---

<sup>1</sup>According to dalvik's bytecode documentation (<http://source.android.com/devices/tech/dalvik/dalvik-bytecode.html>), the source register accepts values between 0 and 65536.

<sup>2</sup>according to the javadoc API

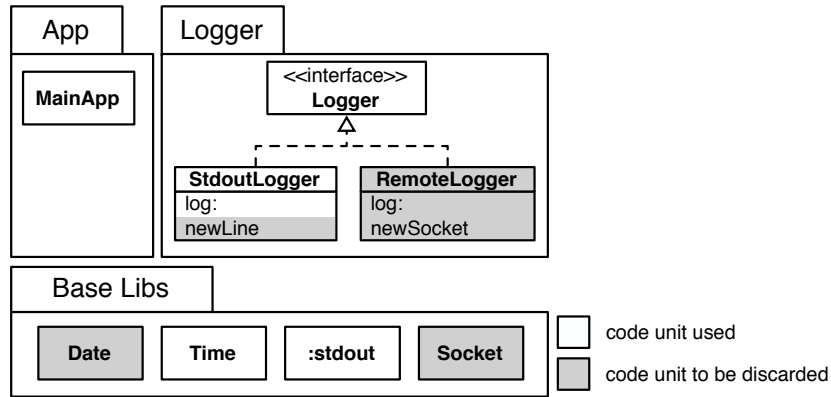


Figure 1: **Example of unused code units.** In gray, the unused code units that can safely be removed.

In this example we can detect the following unused code units, shown in gray in Figure 1 and Figure 2:

1. The logger library includes two logging classes (StdoutLogger and RemoteLogger). Only the StdoutLogger is used and thus, the RemoteLogger class can be discarded.
2. Since the MainApp class does not use the Socket class nor the RemoteLogger class (the only user of the Socket class), the Socket class can be discarded.
3. No class in the application makes use of the Date class. Then, this class can be safely removed.
4. The method newLine (lines 7-8 of Figure 2) of the StdoutLogger class is not used and can be also removed.
5. The StdoutLogger class uses the Time class to print the current time. Then, all code units that are not related to the Time now resolution or printing (*i.e.*, time arithmetic) could be considered as unused.

We would like to generate a new version of this application not containing these unused code units while keeping the application’s behavior. We call this technique *Deployment Unit Tailoring* or *Application Tailoring*.

## 1.2 Challenges of Application Tailoring

A lot of work exists on the tailoring of statically-typed applications [4, 20, 12, 18, 24, 17, 23], where the type annotations aid in the resolution of which piece of code will be used during runtime. However, static analysis is not an option in the context of dynamically-typed languages or in the presence of meta-programming and reflection [11]. In this context of dynamically typed and object-oriented programs that may

```

1  MainApp>>start
2    logger := StdoutLogger new.
3    logger log: 'Application has started'.
4    "do something"
5    logger log: 'Application has finished'.
6
7  StdoutLogger>>newLine
8    stdout newLine.
9
10 StdoutLogger>>log: aMessage
11   stdout nextPutAll: Time now printString.
12   stdout nextPutAll: aMessage.
13   stdout newLine.
14
15 RemoteLogger>>log: aMessage
16   | socket |
17   socket := self newSocket.
18   socket nextPutAll: Time now printString.
19   socket nextPutAll: aMessage.
20   socket newLine.
21
22 RemoteLogger>>newSocket
23   "..."
24   "creates an instance of socket given some configuration"

```

Figure 2: **Code of the unused code units example.** In gray, methods not used by the application.

use reflection, we identify the following main challenges for detecting unused code units:

**Dynamic typing.** Dynamic languages cannot benefit from static analysis due to the absence of type annotations. Those techniques used to detect used code units, such as call-graph analysis, need the support of more dynamic techniques such as tracking runtime information, following the application’s execution flow, or performing symbolic execution.

**Polymorphism and inheritance.** Polymorphism in object-oriented languages allows a code unit to treat objects of different concrete types in the same way as soon as they share a common interface. Inheritance plays a similar role: any class can extend another class and provide different behavior while sharing the same API. As a consequence, both polymorphism and inheritance make the behavior of a program more difficult to predict by just analyzing its code units [22].

**Base libraries are often VM managed.** In most of the modern object-oriented languages, base language libraries such as Java’s bootstrap class loaders or native methods are loaded and initialized by the Virtual Machine (VM) or some low-level component. Since most applications do not use all standard libraries even

if they are initialized, these often big code bases are potentially candidates for removal. However, this raises a challenge since it often requires VM modifications.

**Application runtime configuration.** Modern applications often contain libraries and frameworks besides their proper code. To make these different code units fit together, applications rely on heavy configurations. These configurations are usually present in configuration files looked up dynamically by the application. Based on these configurations, the dependency injection pattern is usually used to dynamically set up the application. This recurrent and standard process for configuring applications implies that static analysis will be inefficient to detect used code units without library-specific knowledge.

**Reflection.** Reflection makes static analysis inoperative by allowing an application to execute unanticipated pieces of code. Any String resulting from a program execution or program configuration can denote a message send<sup>3</sup>, the name of a class to be instantiated, or even a script to be executed. Reflection is indeed important to cover, since it is a broadly used tool in industrial applications with object relational mappers such as Hibernate or Glorp and web frameworks such as Ruby On Rails, Struts or Seaside.

### 1.3 Tornado in a Nutshell.

The rest of this report describes *Tornado*: a solution to these issues with a novel flexible application tailoring technique. Tornado uses a run-fail-grow approach to identify during runtime those code units that are actually used in an application. It consist into "growing" a seed into a deployable specialized version of an application. Missing code units are used to "feed" a *new* minimal version of the application (the *seed*). The resulting deployable application only embeds the seed and used code. By carefully choosing the seed, different level of tailoring are possible. For example, a seed that includes all the base libraries makes the tailoring process to only select used code in the application-specific part; whereas an empty seed makes the tailoring process to select used code in all parts: base libraries, application libraries and application-specific part. The dynamic nature of our solution allows its usage in languages without type annotations. Our solution does not need to modify the original application thanks to its run-fail approach. It also successfully deals with applications that make use of reflection.

## 2 The Run-Fail-Grow Approach

We propose Tornado, a run-fail-grow approach for tailoring. Tornado works by launching a *nurtured* application that has only part of its required code units installed and a *reference* application encompassing all the code units that resulted from the development process. When a failure is detected in the nurtured application, Tornado takes

---

<sup>3</sup>We refer method invocations as message sends because they represent better from our understanding the dynamic property of the invocation.



the missing code units from the reference application and install them into the nurtured application. Thus, the nurtured application grows progressively as failures are found. Once finished, the nurtured application is ready to be deployed on target devices. Figure 3 depicts the basics of our run-fail-grow approach.

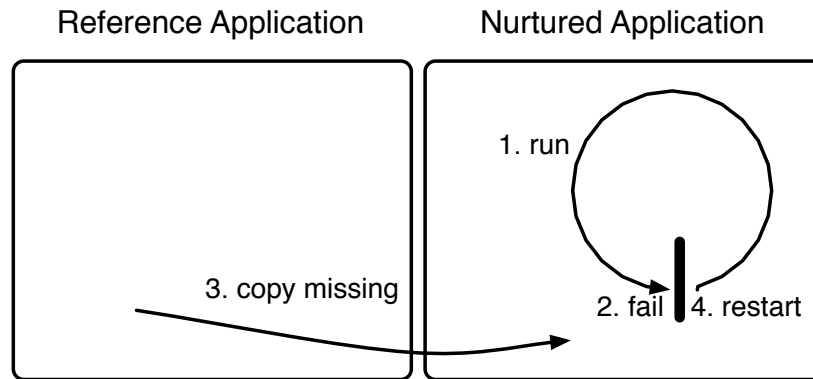


Figure 3: **Application tailoring with a run-fail-grow approach.** We run the nurtured application (1) and detects the missing units on failure (2). At each failure, missing code units are installed from the reference application (3) and the execution restarted (4) until the process finishes.

Tornado starts by launching the reference application to create initial objects and perform startup computations. The reference application is then paused so its state do not change during the tailoring process. Pausing consists in suspending all processes and threads from the application.

Initially, the nurtured application is initialized with only a *seed* embedding code units that developers want to ensure into the deployed application. When the nurtured application starts from a seed that contains the language base libraries, the tailoring will only affect the application specific code units and third-party libraries. When it starts from an empty seed, it will also tailor base libraries.

Following, it installs one or more *application's entry points*. An application's entry point consists in one or more statements that perform some initial computations of the application (*e.g.*, a *main()* method in Java, or the initial method of a thread). The execution of an entry point will result into sending messages to some objects. Required code units will then be cloned on demand from the reference application into the nurtured one. Duplication is performed lazily. For example, when duplicating a class, the content of its fields is not duplicated with it, but deferred until it is actually needed. Also, methods are not duplicated until they are invoked. The process repeats until the user ends it explicitly. Ideally, the nurtured application reaches a stable point where it needs no more code units. The nurtured application is then ready for deployment.

## 2.1 Run-Fail-Grow with an example

We illustrate in this section the ideas behind our tailoring approach with the example introduced in Section 1.1. For the sake of clarity, in this example we will tailor the application's code units and not the base libraries i.e. the seed includes the base libraries.

**Setup of Tornado.** First, Tornado launches reference application with all its code units (cf. Figure 4) and the nurtured application (cf. Figure 5 Step 0). It initializes the nurtured application with a seed containing the language base libraries. Thus, each application has its own copy of the base libraries of the language, as shown in this case with the Date and Time classes and the stdout object.

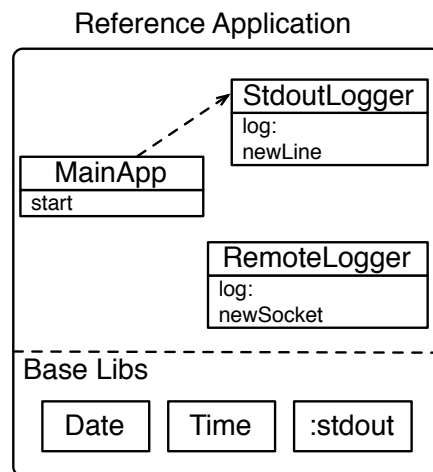


Figure 4: Reference application with all code units.

**Install the application's entry point.** We install the entry point of the application into the nurtured application. In our example, this is an instance of the MainApp class performing the start message (cf. Figure 5 Step 1). Note that although we are referencing an instance of the class MainApp, the MainApp class is not installed yet.

When the mainApp instance receives the start message Tornado realizes that the MainApp class and the start method do not exist in the nurtured application. Tornado then installs these two missing code units (cf. Figure 5 Step 2) and finally the MainApp»start method is activated and starts running.

**Activating the start method.** The execution starts by activating the start method, defined in Figure 2. As we can see in Figure 5 Step 2, the StdoutLogger class does not exist yet in the nurtured application. When it's the turn of the execution of the first statement of the start method (line 2 Figure 2), Tornado captures the message new.

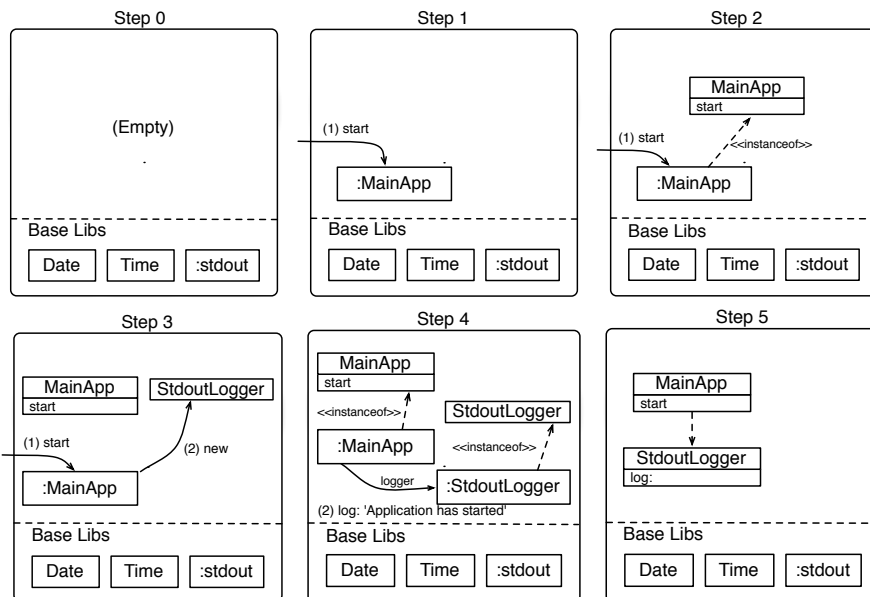


Figure 5: **The nurtured application.** It starts with the base libraries. Application classes are installed on demand.

Thus, before the statement's execution, Tornado installs a StdoutLogger class the same shape as its original counterpart (cf. Figure 5 Step 3). However, it will not contain all the methods nor the meta-data (e.g., superclass, package, subclasses) existing in the reference class since they may not be necessary.

Once Tornado installs the StdoutLogger class, it sends the message `new` to it. This message-send results into a new StdoutLogger instance. Tornado is not involved in the resolution of the new message because this method is part of the language base library, already available in the seed.

The second statement of the `start` method (line 3 Figure 2) is now executed. The logger instance receives the message `log:` with its corresponding argument (cf. Figure 5 Step 4). Tornado captures the `log:` message because the corresponding method is not installed in the StdoutLogger class. Thus, it installs the method inside the corresponding class, and re-sends the message to the logger instance. This time the method is found, and the `log:` method is activated.

Once the `log:` method finishes, the execution returns to the `start` method. There, the third statement (line 5 Figure 2) is executed with no intervention of Tornado, since the `log:` method is already available. Figure 5 Step 5 shows the finally nurtured application: it contains only the methods and classes that are actually used from our application. Leaf objects used during the process were garbage collected.

## 2.2 Tornado’s architecture

Tornado is based on an architecture that allows the complex manipulation of both the nurtured and the reference applications. Examples of such complex manipulations are *e.g.*, Tornado must be notified when a failure occurs in the nurtured application because some code unit is missing, it should be able to start/pause/stop its execution at safe points, install missing code units such as classes and methods, and query runtime information from both the reference and nurtured applications. Our approach performs those tasks through the manipulation of the *object runtime systems* of the applications. An object runtime system is the runtime system of an object-oriented application *e.g.*, a running Java virtual machine executing a Java program, and thus, containing the objects and classes of the application. We identify the following components as a part of the architecture of our solution (cf. Fig. 6):

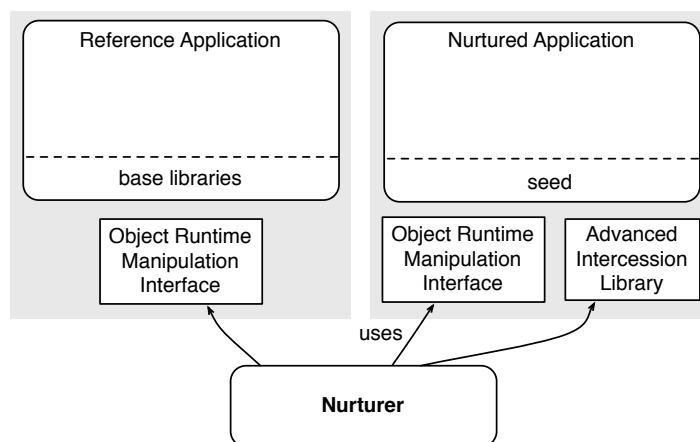


Figure 6: Tornado’s architecture overview.

**Object runtime manipulation interface.** An object runtime manipulation interface allows one to control the runtime execution (starting, pausing and restarting it, and installing new threads/processes), install and load code units such as classes and methods, and query runtime information like the loaded classes, from an object runtime system. A well known example of such an interface is the JVM TI (JVM tool interface) [9]. A typical use of this module is to suspend the execution of the nurtured application while installing missing code units.

**Advanced intercession module.** An advanced intercession module allows advanced reflective capabilities such as modifying an object’s behavior during runtime. Tornado uses this module to capture message sends and so to be notified when it finds missing code units. JRebel [26], Reflectivity [5] or Bifrost [19] are examples of such intercession libraries.

## 2.3 Detecting Missing Code Units with a Run-Fail approach

Tornado gets notified whenever a certain code unit is missing by installing traps in the nurtured application. Traps are installed dynamically following the information flow of the application *e.g.*, when a method A is installed some traps are installed on it before resuming the execution. Tornado works under a run-fail-grow process based on its traps, as shown in Algorithm 1. Whenever a trap is found during execution, Tornado installs the missing code unit, installs some new traps if needed, and finally prepares the application to continue its execution by restarting the message-send that activated the trap.

```
Initialize reference application;
Initialize nurturing application with the seed;
Install entry point(s);
while not finished do
  run the nurtured application;
  if trap was activated then
    install missing code units;
    restart message send;
  end
end
```

**Algorithm 1:** An abstract view of the run-fail-grow process

We identified the following as the basic traps that are necessary to tailor an application:

**Missing object trap.** A *missing object* trap captures message-sends to objects that do not yet exist inside the nurtured application such as classes. When Tornado finds one of these traps, its responsibility is to install the corresponding object. The object installed should be a clone of the original object, containing traps to capture the access to its class, instance variables and fields.

**Missing method trap.** A *missing method* trap captures method invocations whose methods are not defined in the nurtured application yet. When Tornado detects one of these traps, it installs the corresponding method in the class hierarchy of the object. In case some classes are missing, Tornado installs them too. Missing method traps should also capture overridden methods. If an overridden method is not trapped, the method lookup may find a superclass implementation and execute it, resulting into an unexpected behavior. Figure 7 illustrates this problem: the class B from the reference application contains an override, while it is not present in the nurtured application. If no trap is placed to capture the override, the method `doSomething` from class A would be executed, thus changing the semantics of our application.

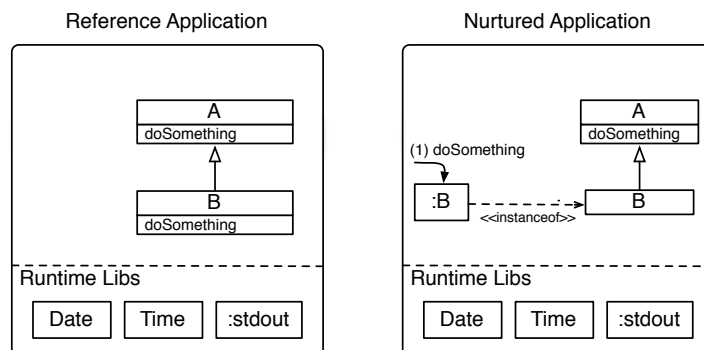


Figure 7: **The need of override traps.** Method traps should capture the overridden doSomething message-send to avoid the superclass method to be executed wrongly.

### 3 Chosen Technology: Pharo

We implemented Tornado in the Pharo programming language. Pharo is a reflective and dynamic programming language inspired on Smalltalk. In Pharo, code units such as classes and methods are reified, allowing their manipulation as any other object in the language. These reifications eased our implementation: classes and methods can be treated in the same way as other terminal objects such as strings with no extra effort.

Pharo is a modern programming language including modern language features and a growing infrastructure including:

**JIT Virtual Machine.** Pharo presents a modern Virtual Machine based on Cog’s architecture [14], containing a Just in Time compiler, polymorphic inline caches and stack optimizations.

**Open classes and class extensions.** Open classes and class extensions [2] allow a package to define methods in classes from other packages.

**Advanced reflection.** Pharo possess the same reflective facilities present originally in Smalltalk-80 [6]. Introspection and intercession in both behavioral and structural axis.

**First Class Instance Variables.** Pharo includes from its 3.0 release first class instance variables [25], namely Slots.

**Traits.** Pharo programming model includes Traits [21] in addition to classes to define the behavior of the system.

## 4 The object runtime manipulation interface: Oz object spaces

Tornado monitors the execution of the nurtured application and manipulates it during runtime as explained in Section 2.2. We built Tornado using our Oz<sup>4</sup> object spaces [15] solution as the *object runtime manipulation interface*. Oz is an extension of the Pharo programming language implementing *protection domains* so called *object spaces*. Oz presents a first class representation of an object space providing a high level API to manipulate those protection domains.

In our Tornado implementation, the reference and nurtured applications are contained each in a different object space. Tornado places traps inside the nurtured object space and starts its execution. This execution is performed directly on a Pharo Virtual Machine, and thus, there is no speed overhead as soon as traps are involved. Whenever the nurtured object space's execution finds a trap, it pauses and returns the control to Tornado. Tornado inspects the classes and methods in the reference object space through mirrors [3] and installs the needed code units from the reference object space on demand, either by creating new objects or compiling new methods. Then, it restarts the nurtured object space's execution from the message send that activated the trap.

## 5 The Execution Traps: Advanced Intercession with Ghost-like Proxies

Implementing execution traps such as the ones described in Section 2.3 requires a powerful intercession module or library. Traps must capture *all* message sends to objects provided by the language runtime as well as the application objects. They must capture *self* and *super* message sends, as well as detect overrides.

To achieve this behavior, we implemented a set of proxies following the Ghost model [13]. Ghost is a low-memory footprint, general purpose proxy model supporting the creation of proxies for normal objects as well as classes and methods. Proxies allow detecting all situations corresponding to our traps. Tornado handles a table relating each proxy to the code unit or object it represents in the reference application. Additionally, each proxy is attached to a *handler* that may perform some action when detecting a message send. We rely on this concept to perform the right action for each trap. We discuss below the different kinds of proxies and handlers we use and how they support our run-fail-grow approach.

**Missing object trap.** This trap is implemented as a proxy taking the place of the original object. This trap is triggered when the proxy receives a message. Its handler replace the proxy by a copy of the original object from the reference application. The replacement of the proxy is achieved through the *become:* facility of the Pharo language that swaps pointers. Each field and the class of this new installed object are installed as new missing object traps.

---

<sup>4</sup>Not related to the Oz programming language. The name Oz for Pharo is inspired on the metaphor of multiple world manipulation.

**Missing method trap.** We implemented the missing method trap in Tornado as a class proxy located at the top of the class hierarchy. Whenever a message is sent to an object, the VM looks up the method in the object's class hierarchy. This trap is triggered whether a message arrives to the top of the hierarchy, meaning that there was no method for it in the hierarchy. When triggered, the handler installs the classes part of the hierarchy of this method and the missing method in its corresponding class. If no method is found to install in the hierarchy of the reference object space, Tornado sends the `doesNotUnderstand:` message (an equivalent to *e.g.*, Ruby's `method_missing` and Python's `__getattr__`) to honor the dynamic semantics of Pharo.

**Missing override trap.** We implemented missing override traps in Tornado using method proxies. Method proxies are placed in the method dictionaries of classes containing overridden methods, taking the place of the original method. When Tornado installs a class into the nurtured application that contains overridden methods in the reference object space, it installs into this class a method proxy for each of its overridden methods. This proxy is activated whenever it is about to be executed. The handler of this trap takes the original method from the reference object space and compiles a new one with its same source code inside the nurtured object space.

**Primitive methods trap.** *Primitive method* traps are implementation specific related to the Pharo language. Pharo's primitive operations such as number arithmetic are implemented through primitive methods. Primitive methods are implemented in the Virtual Machine and do often access directly the fields of its receiver and arguments by forging references and manipulating directly the memory. Thus, when a *missing object trap* proxy is the argument of such a method the VM can modify this proxy without activating the trap. Primitive method traps decorate Pharo's primitive methods to capture their execution and trigger each of the missing object traps received as arguments. In this way, Tornado forces the installation of these traps and the primitive is executed with actual objects instead of proxies, as expected.

## 6 Object Installation and Object Mappings

Tornado installs all objects inside the nurtured application on demand. When Tornado installs an object inside the nurtured application, this new object has the same format and size as its original counterpart. *Propagation rules* determine how each of the object's fields are treated on installation. Tornado provides the following propagation rules to customize installation:

**Missing object trap.** This is the default propagation rule. This propagation rule installs a missing object trap in each field of the object that is being installed.

**Materialization.** This propagation rule forces the installation of the object referenced by the field. This is used for those cases where some structure should be guaranteed to the Virtual Machine.



**Swapping.** This propagation rule forces the reference of the object installed be swapped to another object's reference. The usual use case of this rule is to replace the reference to a reference to the nil object, and so, force lazy initializations.

Tornado takes care of the identity of objects with an identity table. This is important because Tornado works at the object granularity. Due to the inherent graph nature of object-oriented programs, an object being installed may reference another object that is already installed inside the nurtured application. In such a case, Tornado ensuring the correctness of the graph.

End user applications can be tailored usually with the default propagation rules. However, other propagation rules serve to the purpose of tailoring the base libraries. In Pharo, some special objects are used by the Virtual Machine (VM) and their installation should be forced to ensure its correct behavior. For example, the first three fields of class objects (superclass, format and method dictionary) cannot be proxified because they are used by the VM for the method lookup. The same happens with other objects reifying low level concepts such as methods, activation records or semaphores.

## 7 Handling Tailoring Levels through Seeds

Tornado's seeds specify the level of tailoring. The seeds are in charge of initializing the nurtured application's object space with the elements we want to ensure on it. Note that a seed can indeed contain any arbitrary code units and objects. They are not restricted to have only base or third party libraries. The selection or extraction of what is included as part of a seed is application dependent and orthogonal to the run-fail-grow process. Our current prototype supports two ways of describing and building seeds:

### 7.1 Loading an already existing memory snapshot

The nurtured application's object space is initialized by loading an already existing snapshot or image (*i.e.*, this is an image in the same sense as Smalltalk or Lisp). This technique consists in using a memory dump from an object heap containing all the classes and objects desired in the seed. This memory snapshot should follow Pharo's object format.

Oz provides a primitive method to do such image loading. An image file is read, the internal representation of an object space is initialized, and an object space is retrieved.

### 7.2 Creating all seed code units from scratch

The nurtured application's object space is initialized with objects built from scratch. This technique uses a bootstrapping process we defined in [16]. With bootstrapping, we describe declaratively the contents we want in the seed and we build it automatically.

## 8 Preparing the Application for Deployment

Once Tornado finishes and the application contains all the code units needed for running, it prepares it for deployment. That is: removing all trap leftovers and extract the nurtured application.

To remove the trap leftovers, Tornado identifies the traps by the presence of proxies and replaces the references to those proxies by references to another object, defaulting to the nil object. Proxy objects do not represent a drawback in space consumption because they are garbage collected. Once the traps are removed, the nurtured application keeps no dependencies to Tornado nor its infrastructure. Thus, the application can run outside the Oz infrastructure with no performance penalties.

### 8.1 Snapshot

To extract the nurtured application and allow it to run outside of Tornado's infrastructure, our implementation makes a snapshot of it: it saves all objects, classes and methods in a binary format file to be later started with the actual state. Making an snapshot pays off because it speeds up the loading and starting time of the application. The snapshot technique is the same used by image-based languages such as Smalltalk, Lisp.

### 8.2 Building a Static Description

Alternatively to the snapshot approach, Oz allows one to inspect the state of an object space to know which classes and methods are installed inside it. Therefore we can build an static representation of the nurtured application code units. Appendix A, Appendix C and Appendix D shows the list of methods tailored by tornado in our case studies, extracted using Oz.

## 9 Results

### 9.1 Methodology

We tested our Tornado implementation by tailoring two different Pharo applications: a hello world application and a simple but yet interactive web application based on the Seaside framework [1]. Our methodology consisted in: setting up a seed for the application, preparing the application entry points and executing the application. In the case of the interactive web application, we interacted with it through a web browser. Once we finished the process, we extracted the resulting application by making a snapshot of it in a Pharo image file. We tested the generated snapshots to verify they work properly (under the assumption that only the previously used features of the application should work).

Finally, to present our results we measured the size of the generated snapshots files and compared them with the snapshots of the full applications under Pharo's production

option<sup>5</sup>. The results prove the soundness of our solution.

## 9.2 Hello World Application

We used Tornado to tailor a *hello world* application writing 10 times the ‘hello world’ string to the standard output (stdout). In this case study we used an empty seed to grow both base libraries and the application’s code. Figure 8 shows the installed entry point to tailor this application. Table 1 shows our results for this case. We succeed to reduce the application’s size to 1% of its original counterpart.

```
1 FileStream startUp: true.  
2 1 to: 10 do: [:i | FileStream stdout nextPutAll: 'hello'; crlf].
```

Figure 8: Entry point of the Hello World application with an empty seed.

	Size(KB)	Occupied(%)	Saved(%)
Reference	12872	100%	0%
Tailored	131	1%	99%

Table 1: Results of the tailored Hello World application.

## 9.3 Seaside Web Application

We also used Tornado to tailor a simple web application consisting in a webpage with a counter containing two buttons. These two buttons perform requests to the web server to increase and decrease the counter. The Seaside application framework was configured with its default values, without making any customizations.

In this case, we used two different seeds for tailoring: a seed containing all Pharo base libraries and an empty seed. Appendix B presents the entry points for these both seeds. The tailoring was done by starting the application and exercising it by generating requests through a web browser, clicking on its decrease and increase buttons.

Table 2 shows the results obtained when tailoring this application with each of these two seeds. Figure 9 presents a tailoring map illustrating how Tornado selects the code units from a reference application given a seeds. This figure also presents the notation we use in Table 2:  $P$  is the Pharo base libraries,  $S$  is the Seaside Framework and  $C$  is the Counter application code units present in the reference application.  $P'$ ,  $S'$  and  $C'$  are their counterparts selected by Tornado when using an empty seed.  $P''$ ,  $S''$  and  $C''$  are their counterparts, as selected by Tornado when using a seed with all base libraries. In the latter, we can note that  $P=P''$ .

<sup>5</sup>Pharo allows to prepare a snapshot for production. This option cleans some caches and removes some well known objects from the system, thus, freeing space.

	Size (KB)		
Ref. Pharo Base Libraries ( $P$ )			12872
Ref. Seaside Framework ( $S$ )			4326
Ref. Counter Application ( $R$ )			52
<b>Total Ref. Application (<math>P+S+C</math>)</b>			<b>17250</b>
	Size (KB)	Occupied (%)	Saved (%)
$P'+S'+C' / P+S+C$	573	3%	97%
$P''+S''+C'' / P+S+C$	13090	76%	24%
$S''+C'' / S+C$	218	5%	95%

Table 2: **Results of second case study.** Results of tailoring a web application with two different seeds. On the left, the total sizes of the original application deployment components (base libraries, application framework and counter application). On the right, our results when applying after tailoring. The first two results rows are compared against the total of the reference application. The third row presents the comparison without including base libraries, already inside the seed.

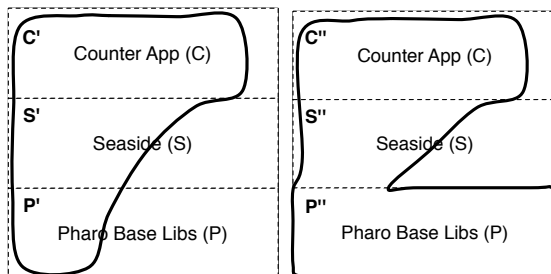


Figure 9: **Tailoring Map.** Tailoring map describing the Seaside application generated with the empty seed (left) and the full Pharo seed (right).

## 10 Conclusion

In this paper our run-fail-grow approach for tailoring applications, namely Tornado. We presented both a reference model and a prototype implementation. Tornado tailors an application by starting it with a minimal set of entry points and installing missing code units on demand, following the execution flow of the application. Tornado defines a series of traps to detect missing code units.

By following the runtime execution, it supports dynamic features such as reflection and meta-programming. Tornado is able to tailor third party libraries and base libraries of the language, and so, to produce minimal footprint deployment units for applications. We validate our approach on two different applications: an hello world and a Web application. The results show that memory has been successfully saved while the resulting applications do still work.

We identify as future work the dynamic adaptation of a distributed Tornado-like system. We believe that our model in conjunction with reflective language architectures are one more step closer of the runtime adaptation and distribution of applications with

almost zero downtime.

**Acknowledgements.** This work was supported by Ministry of Higher Education and Research, Nord-Pas de Calais Regional Council, FEDER via the 'Contrat de Projets Etat Region (CPER) 2007-2013', the Cutter ANR project, ANR-10-BLAN-0219.

## References

- [1] A. Bergel, S. Ducasse, and L. Renggli. Seaside – advanced composition and control flow for dynamic web applications. *ERCIM News*, 2008.
- [2] A. Bergel, S. Ducasse, and R. Wuyts. Classboxes: A minimal module model supporting local rebinding. In *Proceedings of Joint Modular Languages Conference (JMLC'03)*, volume 2789 of *LNCS*, pages 122–131. Springer-Verlag, 2003. Best Paper Award.
- [3] G. Bracha and D. Ungar. Mirrors: design principles for meta-level facilities of object-oriented programming languages. In *OOPSLA'04*, 2004.
- [4] A. Courbot, J.-J. Vandewalle, and G. Grimaud. Romization: Early deployment and customization of Java systems for constrained devices. In *CASSIS'05*, 2005.
- [5] M. Denker. *Sub-method Structural and Behavioral Reflection*. PhD thesis, University of Bern, May 2008.
- [6] B. Foote and R. E. Johnson. Reflective facilities in Smalltalk-80. In *Proceedings OOPSLA '89, ACM SIGPLAN Notices*, volume 24, pages 327–336, Oct. 1989.
- [7] D. Grove, G. DeFouw, J. Dean, and C. Chambers. Call graph construction in object-oriented languages. In *OOPSLA '97*, 1997.
- [8] Java micro edition. <http://java.sun.com/javame/index.jsp>.
- [9] Sun microsystems, inc. JVM tool interface (JVMTI). <http://java.sun.com/j2se/1.5.0/docs/guide/jvmti/>.
- [10] S. Liang and G. Bracha. Dynamic class loading in the Java virtual machine. In *OOPSLA'98*, 1998.
- [11] M. P. Mariano. *Application-Level Virtual Memory for Object-Oriented Systems*. PhD thesis, Université de Lille, 2012.
- [12] K. Marquet, A. Courbot, and G. Grimaud. Ahead of time deployment in rom of a java-os. In *ICESS'05*, 2005.
- [13] M. Martinez Peck, N. Bouraqadi, M. Denker, S. Ducasse, and L. Fabresse. Efficient proxies in Smalltalk. In *IWST'11*, 2011.
- [14] E. Miranda. The cog smalltalk virtual machine. In *Proceedings of VMIL 2011*, 2011.
- [15] G. Polito, S. Ducasse, L. Fabresse, and N. Bouraqadi. Virtual smalltalk images: Model and applications. In *IWST - International Workshop on Smalltalk Technology, Co-located within the 21th International Smalltalk Conference - 2013*, 2013.
- [16] G. Polito, S. Ducasse, L. Fabresse, N. Bouraqadi, and B. van Ryseghem. Bootstrapping reflective systems: The case of pharo. *Science of Computer Programming*, 2013.
- [17] L. Popa, C. Raiciu, R. Teodorescu, I. Athanasiu, and R. Pandey. Using code collection to support large applications on mobile devices. In *MobiCom '04*, 2004.
- [18] D. Rayside and K. Kontogiannis. Extracting java library subsets for deployment on embedded systems. *Sci. Comput. Program.*, 2002.

- [19] J. Ressia. *Object-Centric Reflection*. PhD thesis, Institut für Informatik und angewandte Mathematik, 2012.
- [20] C. Rippert, A. Courbot, and G. Grimaud. A low-footprint class loading mechanism for embedded java virtual machines. In *PPPJ '04*, 2004.
- [21] N. Schärli, S. Ducasse, O. Nierstrasz, and A. P. Black. Traits: Composable units of behavior. Technical Report IAM-02-005, Institut für Informatik, Universität Bern, Switzerland, Nov. 2002. Also available as Technical Report CSE-02-014, OGI School of Science & Engineering, Beaverton, Oregon, USA.
- [22] D. Taenzer, M. Ganti, and S. Podar. Problems in object-oriented software reuse. In *ECOOP'89*, 1989.
- [23] R. Teodorescu and R. Pandey. Using jit compilation and configurable runtime systems for efficient deployment of java programs on ubiquitous devices. In *UbiComp '01*, 2001.
- [24] F. Tip, P. F. Sweeney, and C. Laffra. Extracting library-based java applications. *Commun. ACM*, 2003.
- [25] T. Verwaest, C. Bruni, M. Lungu, and O. Nierstrasz. Flexible object layouts: enabling lightweight language extensions by intercepting slot access. In *Proceedings of 26th International Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA '11)*, pages 959–972, New York, NY, USA, 2011. ACM.
- [26] ZeroTurnAround. What developers want: The end of application redeploys. <http://files.zereturnaround.com/pdf/JRebelWhitePaper2012-1.pdf>, 2012.

## A Appendix: Method List of a Nurtured Hello World Application

List of methods extracted from the nurtured Hello World application. This list includes all methods installed from the Pharo base libraries and the simple Hello World application.

Array class»new:	FileStream»openOnHandle:name:forWrite:Standard
ArrayedCollection»size	FileStream»primWrite:from:startingAt:count:Standard
Association class»key:value:	GreekEnvironment class»supportedLanguages
Association»value:	HashTableSizes class»atLeast:
Association»value	HashTableSizes class»sizes
BlockClosure»on:do:	HashedCollection class»newProto
BlockClosure»repeat	HashedCollection»atNewIndex:put:
BlockClosure»valueNoContextSwitch	HashedCollection»findElementOrNil:
ByteString class»compare:with:collated:	HashedCollection»fullCheck
ByteString class»findFirstInString:inSet:startingAt:	HashedCollection»grow
ByteString class»stringHash:initialHash:	HashedCollection»initialize:
ByteString»at:put:	Integer»asCharacter
ByteString»at:	JapaneseEnvironment class»supportedLanguages
ByteString»isByteString	KoreanEnvironment class»supportedLanguages
ByteString»replaceFrom:to:with:startingAt:	LanguageEnvironment class»currentPlatform
ByteTextConverter class»unicodeToByteTable	LanguageEnvironment class»defaultSystemConverter
ByteTextConverter»nextPut:toStream:	LanguageEnvironment class»initKnownEnvironments
ByteTextConverter»unicodeToByte:	LanguageEnvironment class»knownEnvironments
Character class»cr	LanguageEnvironment class»localeID:
Character class»lf	LanguageEnvironment»localeID:
Character class»value:	Latin1Environment class»supportedLanguages
Character»=	Latin2Environment class»supportedLanguages
Character»asInteger	Latin9Environment class»supportedLanguages
Character»asciiValue	Latin9Environment class»systemConverterClass
Character»charCode	Locale class»currentPlatform
Collection»detect:ifNone:	Locale class»determineCurrentLocale
Dictionary»at:ifAbsent:	Locale»determineLocaleID
Dictionary»at:ifPresent:	Locale»determineLocale
Dictionary»at:put:	Locale»fetchISO2Language
Dictionary»noCheckAdd:	Locale»languageEnvironment
Dictionary»scanFor:	Locale»localeID:
FileStream class»newForStdio	Locale»localeID
FileStream class»new	Locale»primCountry
FileStream class»standardIOStreamNamed:forWrite	Locale»primLanguage
FileStream class»startUp:	LocaleID class»isoLanguage:isoCountry:
FileStream class»stdioHandles	LocaleID class»isoLanguage:
FileStream class»stdoutCharacter»isCharacter	LocaleID class»isoString:
FileStream class»voidStdioFiles	LocaleID»=
FileStream»collectionSpeciesStandard	LocaleID»hash
FileStream»enableReadBufferingSmalltalkImage»vm	LocaleID»isoCountry
FileStream»isBinaryStandard	LocaleID»isoLanguage:isoCountry:
FileStream»next:putAll:startingAt:Standard	LocaleID»isoLanguage
FileStream»nextPut:Standard	LookupKey class»key:

LookupKey»key:	Semaphore»critical:
LookupKey»key	SequenceableCollection»copyFrom:to:
Magnitude»max:	SequenceableCollection»copyUpTo:
MultiByteFileStream»basicNext:putAll:startingAt:	SequenceableCollection»do:
MultiByteFileStream»basicNextPut:	SequenceableCollection»first:
MultiByteFileStream»converter:	SequenceableCollection»first
MultiByteFileStream»converter	SequenceableCollection»identityIndexOf:ifAbsent:
MultiByteFileStream»installLineEndConventionInCornerRadius:	SequenceableCollection»identityIndexOf:
MultiByteFileStream»nextPut:	SequenceableCollection»indexOf:ifAbsent:
MultiByteFileStream»nextPutAll:	SequenceableCollection»second
Number»negative	SequenceableCollection»writeStream
OSPlatform class»isWin32	SimplifiedChineseEnvironment class»supportedLanguages
OSPlatform class»platformName	SmallInteger»bitXor:
Object»=	SmallInteger»hash
Object»at:put:Object»isCharacter	Stream»basicNextPut:
Object»at:	String class»new:
Object»class	String class»with:
Object»hash	String»=
Object»isInteger	String»compare:with:collated:
Object»species	String»findDelimiters:startingAt:
Object»~~	String»findTokens:
OrderedCollection class»arrayType	String»hash
OrderedCollection class»new:	String»indexOf:startingAt:ifAbsent:
OrderedCollection class»new	String»isString
OrderedCollection»add:	String»skipDelimiters:startingAt:
OrderedCollection»addLast:	TextConverter class»defaultSystemConverter
OrderedCollection»at:	TextConverter class»initializeLatin1 MapAndEncodings
OrderedCollection»ensureBoundsFrom:to:	TextConverter class»latin1 Encodings
OrderedCollection»resetTo:	TextConverter class»latin1 Map
OrderedCollection»reset	TextConverter»initialize
OrderedCollection»setCollection:	TextConverter»installLineEndConvention:
OrderedCollection»size	TextConverter»nextPutAll:toStream:
PositionableStream class»on:	TextConverter»nextPutByteString:toStream:
PositionableStream»isBinary	VirtualMachine class»getSystemAttribute:
PositionableStream»on:	WriteStream»contentsStandard
ProtoObject»basicIdentityHash	WriteStream»CrLf
ProtoObject»flag:	WriteStream»nextPut:
ProtoObject»identityHash	WriteStream»on:
ProtoObject»initialize	WriteStream»reset
RussianEnvironment class»supportedLanguages	



## B Appendix: Entry Points to Tailor the Seaside Web Application

Entry points as used to tailor the Seaside web application with a full Pharo seed and an empty seed. The first one (Figure 10) only consists in starting the web server as the base libraries are initialized and available in the seed. The latter one (Figure 11) includes the initialization of the minimal runtime needed to do networking.

```
1 ZnZincServerAdaptor startOn: 8888.
```

Figure 10: Entry point of the Seaside application with a full Pharo seed.

```
1 "We initialize some classes of the system"
2 SmalltalkImage initializeForTornado.
3 Symbol initializeForTornado.
4 Object initialize.
5 ExternalSemaphoreTable initialize.
6 Socket initialize.
7 Delay initialize.
8 Delay startUp: true.
9 Delay shutDown: true.
10 OSPlatform initialize.
11 DiskStore initialize.
12 FileStream initialize.
13 NetNameResolver initialize.
14 DateAndTime initialize.
15 ProcessorScheduler initialize.
16 WeakFinalizationList initialize.
17 UUIDGenerator initialize.
18 WeakArray initialize.
19 GRPharoRandomProvider initialize.
20 WASlime initialize.
21 UIManager basicDefault: DummyUIManager new.
22 ZnServer initialize.
23 WAServerManager initialize.
24 Smalltalk instVarNamed: 'session' put: Smalltalk newSessionObject.
25 Smalltalk startupImage: true snapshotWorked: true.
26
27 "Finally we start the web server"
28 ZnZincServerAdaptor startOn: 8888.
```

Figure 11: Entry point of the Seaside application with an empty seed.

## C Appendix: Method List of Seaside Counter Application with Full Pharo Seed

List of methods extracted from the nurtured Web application when using a seed containing all base libraries from Pharo. This list includes all methods installed from Seaside framework and the counter application. The list of methods part of the base library are excluded as it is the same list of the methods found in Pharo base library.

WAAccessIntervalReapingStrategy»defaultConfiguration	WAAccessIntervalReapingStrategy»initializeWithKey:
WAAccessIntervalReapingStrategy»initialize	WAAccessIntervalReapingStrategy»isAttributeInheritedOn:
WAAccessIntervalReapingStrategy»interval	WAAccessIntervalReapingStrategy»isAttributeLocalOn:
WAAccessIntervalReapingStrategy»reap	WAAccessIntervalReapingStrategy»key
WAAccessIntervalReapingStrategy»stored:key:	WABrush»initialize
WAActionCallback»block:	WABrush»parent
WAActionCallback»evaluateWithArgument:	WABrush»setParent:canvas:
WAActionCallback»isEnabledFor:	WABrush»with:
WAActionCallback»signalRenderNotification	WABufferedResponse class»on:
WAActionPhaseContinuation»continue	WABufferedResponse»contents
WAActionPhaseContinuation»handleRequest	WABufferedResponse»destroy
WAActionPhaseContinuation»renderContext:	WABufferedResponse»initializeOn:
WAActionPhaseContinuation»renderContext	WABufferedResponse»stream
WAActionPhaseContinuation»runCallbacks	WACache»at:ifAbsent:
WAActionPhaseContinuation»shouldRedirect	WACache»expiryPolicy
WAdmin class»defaultServerManager	WACache»initializeCollections
WAdmin class»serverAdaptors	WACache»initializeMutex
WAnchorTag»callback:	WACache»initialize
WAnchorTag»tag	WACache»keyAtValue:ifAbsent:
WAnchorTag»url	WACache»keyAtValue:
WAnchorTag»with:	WACache»keySize
WApplication»contentType	WACache»missStrategy
WApplication»doesHandlerSupportCookies:	WACache»notifyRemoved:key:
WApplication»handleDefault:	WACache»notifyRetrieved:key:
WApplication»handleFiltered:	WACache»notifyStored:key:
WApplication»isApplication	WACache»pluginsDo:
WApplication»isImplemented:	WACache»reapingStrategy
WApplication»keyField	WACache»reap
WApplication»libraries	WACache»removalAction
WApplication»mainClass	WACache»setExpiryPolicy:
WApplication»mimeType	WACache»setMissStrategy:
WApplication»newSession	WACache»setReapingStrategy:
WApplication»resourceBaseUrl	WACache»setRemovalAction:
WApplication»sessionClass	WACache»store:
WApplicationConfiguration»parents	WACacheCapacityConfiguration»describeOn:
WAttributeSearchContext class»key:target:	WACacheMissStrategy»missed:
WAttributeSearchContext»at:ifPresent:	WACachePlugin»configuration
WAttributeSearchContext»at:put:	WACachePlugin»defaultConfiguration
WAttributeSearchContext»attribute	WACachePlugin»initialize
WAttributeSearchContext»cachedValues	WACachePlugin»removed:key:
WAttributeSearchContext»findAttributeAndSelectAndStore:	WACachePlugin»retrieved:key:

WACachePlugin»setCache:  
 WACachePlugin»stored:key:  
 WACacheReapingStrategy»reap  
 WACallback class»on:  
 WACallback»convertKey:  
 WACallback»evaluateWithFieldValues:  
 WACallback»key  
 WACallback»setKey:callbacks:  
 WACallback»valueForField:  
 WACallbackRegistry»advanceKey  
 WACallbackRegistry»handle:  
 WACallbackRegistry»increaseKey  
 WACallbackRegistry»initialize  
 WACallbackRegistry»nextKey  
 WACallbackRegistry»store:  
 WACanvas»brush:  
 WACanvas»flush  
 WACanvas»nest:  
 WACanvas»render:  
 WACanvas»text:  
 WAComponent»accept:  
 WAComponent»acceptDecorated:  
 WAComponent»decoration  
 WAComponent»initialize  
 WAComponent»updateStates:  
 WAConfigurationDescription»add.to:  
 WAConfigurationDescription»addAttribute:  
 WAConfigurationDescription»attributes  
 WAConfigurationDescription»expressions  
 WAConfigurationDescription»initialize  
 WAConfigurationDescription»integer:  
 WAConfiguredRequestFilter»configuration  
 WACounter»count:  
 WACounter»decrease  
 WACounter»increase  
 WACounter»initialize  
 WACounter»renderContentOn:  
 WACounter»states  
 WADefaultScriptGenerator»close:on:  
 WADefaultScriptGenerator»open:on:  
 WADevelopmentConfiguration»parents  
 WADispatcher class»default  
 WADispatcher»handleFiltered:named:  
 WADispatcher»handleFiltered:  
 WADispatcher»handlerAt:ifAbsent:  
 WADispatcher»handlerAt:with:  
 WADispatcher»handlers  
 WADispatcher»nameOfHandler:  
 WADispatcher»urlFor:  
 WADocument class»on:codec:  
 WADocument»close  
 WADocument»destroy  
 WADocument»initializeWithStream:codec:  
 WADocument»nextPut:  
 WADocument»nextPutAll:  
 WADocument»open:  
 WADynamicVariable class»use:during:  
 WADynamicVariable class»value  
 WAEncoder class»on:table:  
 WAEncoder class»on:  
 WAEncoder»initializeOn:table:  
 WAEncoder»nextPut:  
 WAErrorHandler class»exceptionSelector  
 WAExampleComponent»rendererClass  
 WAExceptionFilter»exceptionHandler  
 WAExceptionFilter»handleFiltered:  
 WAExceptionHandler class»context:  
 WAExceptionHandler class»exceptionSelector  
 WAExceptionHandler class»handleExceptionsDuring:context:  
 WAExceptionHandler class»handles:  
 WAExceptionHandler»handleExceptionsDuring:  
 WAExceptionHandler»handles:  
 WAExceptionHandler»initializeWithContext:  
 WAHeaderFields»checkValue:  
 WAHeaderFields»privateAt:put:  
 WAHeadingTag»initialize  
 WAHeadingTag»level1  
 WAHeadingTag»level  
 WAHeadingTag»tag  
 WAHtmlAttributes»encodeOn:  
 WAHtmlAttributes»privateAt:put:  
 WAHtmlCanvas»anchor  
 WAHtmlCanvas»heading:  
 WAHtmlCanvas»heading  
 WAHtmlCanvas»spaceEntity  
 WAHtmlDocument»scriptGenerator:  
 WAHtmlDocument»scriptGenerator  
 WAHtmlElement class»root:  
 WAHtmlElement»attributeAt:put:  
 WAHtmlElement»attributes  
 WAHtmlElement»encodeBeforeOn:  
 WAHtmlElement»encodeOn:  
 WAHtmlElement»initializeWithRoot:  
 WAHtmlElement»isClosed  
 WAHtmlRoot»add:  
 WAHtmlRoot»beXhtml10Strict  
 WAHtmlRoot»bodyAttributes  
 WAHtmlRoot»closeOn:  
 WAHtmlRoot»docType:  
 WAHtmlRoot»htmlAttributes

WAHtmlRoot»initialize	WAMimeType»greaseString
WAHtmlRoot»meta	WAMimeType»main:
WAHtmlRoot»openOn:	WAMimeType»main
WAHtmlRoot»title:	WAMimeType»parameters
WAHtmlRoot»writeElementsOn:	WAMimeType»sub:
WAHtmlRoot»writeFootOn:	WAMimeType»sub
WAHtmlRoot»writeHeadOn:	WAMutex»critical:
WAHtmlRoot»writeScriptsOn:	WAMutex»initialize
WAHtmlRoot»writeStylesOn:	WAMutualExclusionFilter»handleFiltered:
WAHttpVersion class»fromString:	WAMutualExclusionFilter»initialize
WAHttpVersion class»major:minor:	WAMutualExclusionFilter»shouldTerminate:
WAHttpVersion class»readFrom:	WANotifyRemovalAction»removed:key:
WAHttpVersion»initializeWithMajor:minor:	WAOBJECT»application
WAInitialRequestVisitor class»request:	WAOBJECT»requestContext
WAInitialRequestVisitor»initializeWithRequest:	WAOBJECT»session
WAInitialRequestVisitor»request	WAPainter»renderWithContext:
WAInitialRequestVisitor»visitPresenter:	WAPainter»updateRoot:
WAKeyGenerator class»current	WAPainter»updateUrl:
WAKeyGenerator»keyOfLength:	WAPainterVisitor»visitComponent:
WALastAccessExpiryPolicy»defaultConfiguration	WAPainterVisitor»visitDecorationsOfComponent:
WALastAccessExpiryPolicy»initialize	WAPainterVisitor»visitPainter:
WALastAccessExpiryPolicy»isExpired:key:	WAPainterVisitor»visitPresenter:
WALastAccessExpiryPolicy»retrieved:key:	WAPathConsumer class»path:
WALastAccessExpiryPolicy»stored:key:	WAPathConsumer»atEnd
WALastAccessExpiryPolicy»timeout	WAPathConsumer»initializeWith:
WALeastRecentlyUsedExpiryPolicy»defaultConfiguration	WAPathConsumer»next
WALeastRecentlyUsedExpiryPolicy»initialize	WAPathConsumer»upToEnd
WALeastRecentlyUsedExpiryPolicy»isExpired:key:	WAPresenter»childrenDo:
WALeastRecentlyUsedExpiryPolicy»maximumAge	WAPresenter»children
WALeastRecentlyUsedExpiryPolicy»removed:key:	WAPresenter»initialRequest:
WALeastRecentlyUsedExpiryPolicy»retrieved:key:	WAPresenter»script
WALeastRecentlyUsedExpiryPolicy»stored:key:	WAPresenter»style
WAMergedRequestFields class»on:	WAPresenter»updateRoot:
WAMergedRequestFields»allAt:	WAPresenter»updateStates:
WAMergedRequestFields»at:ifAbsent:	WAPresenterGuide class»client:
WAMergedRequestFields»includesKey:	WAPresenterGuide»client
WAMergedRequestFields»initializeOn:	WAPresenterGuide»initializeWithClient:
WAMergedRequestFields»keysAndValuesDo:	WAPresenterGuide»visit:
WAMergedRequestFields»keysDo:	WAPresenterGuide»visitPainter:
WAMetaElement»content:	WRegistryConfiguration»parents
WAMetaElement»contentScriptType:	WRenderContext»actionBaseUrl:
WAMetaElement»contentType:	WRenderContext»actionUrl:
WAMetaElement»encodeBeforeOn:	WRenderContext»actionUrl
WAMetaElement»responseHeaderName:	WRenderContext»callbacks
WAMetaElement»tag	WRenderContext»defaultVisitor
WAMimeType class»fromString:	WRenderContext»destroy
WAMimeType class»main:sub:	WRenderContext»document:
WAMimeType class»text.Javascript	WRenderContext»document
WAMimeType class»text.Plain	WRenderContext»initialize
WAMimeType»charset:	WRenderContext»resourceUrl:

WARenderContext»visitor:	WARRequest»setHeaders:
WARenderContext»visitor	WARRequest»setPostFields:
WARenderLoopConfiguration»parents	WARRequest»setRemoteAddress:
WARenderLoopContinuation»createActionContinuation:	WARRequest»uri
WARenderLoopContinuation»createRenderContinuation:	WARRequest»url
WARenderLoopContinuation»presenter	WARRequestContext class»request:response:codec:
WARenderLoopContinuation»toPresenterSendRoot:	WARRequestContext»application
WARenderLoopContinuation»updateRoot:	WARRequestContext»charSet
WARenderLoopContinuation»updateStates:	WARRequestContext»codec
WARenderLoopContinuation»updateUrl:	WARRequestContext»consumer
WARenderLoopContinuation»withNotificationHandler:	WARRequestContext»destroy
WARenderLoopMain»createRoot	WARRequestContext»handlers
WARenderLoopMain»prepareRoot:	WARRequestContext»handler
WARenderLoopMain»rootClass	WARRequestContext»initializeWithRequest:response:codec:
WARenderLoopMain»rootDecorationClasses	WARRequestContext»newDocument
WARenderLoopMain»start	WARRequestContext»push:during:
WARenderPhaseContinuation»createHtmlRootWithContent:	WARRequestContext»registry
WARenderPhaseContinuation»createRenderContext:	WARRequestContext»request
WARenderPhaseContinuation»handleRequest	WARRequestContext»respond:
WARenderPhaseContinuation»processRendering:	WARRequestContext»respond
WARenderVisitor class»context:	WARRequestContext»responseGenerator
WARenderVisitor»initializeWithContext:	WARRequestContext»response
WARenderVisitor»renderContext	WARRequestContext»session
WARenderVisitor»visitPainter:	WARRequestFilter»handleFiltered:
WARenderer class»context:	WARRequestFilter»initialize
WARenderer»actionUrl	WARRequestFilter»next
WARenderer»callbacks	WARRequestFilter»setNext:
WARenderer»context	WARRequestFilter»updateStates:
WARenderer»document	WARRequestHandler»addFilter:
WARenderer»flush	WARRequestHandler»addFilterLast:
WARenderer»initializeWithContext:	WARRequestHandler»basicUrl
WARenderer»render:	WARRequestHandler»configuration:
WARenderer»text:	WARRequestHandler»configuration
WARRequest class»method:uri:version:	WARRequestHandler»defaultConfiguration
WARRequest»at:ifAbsent:	WARRequestHandler»documentClass
WARRequest»cookiesAt:	WARRequestHandler»filters
WARRequest»cookies	WARRequestHandler»filter
WARRequest»destroy	WARRequestHandler»handle:
WARRequest»fields	WARRequestHandler»initialize
WARRequest»headerAt:ifAbsent:	WARRequestHandler»isApplication
WARRequest»headerAt:	WARRequestHandler»isRoot
WARRequest»initializeWithMethod:uri:version:	WARRequestHandler»parent
WARRequest»isGet	WARRequestHandler»preferenceAt:
WARRequest»isPrefetch	WARRequestHandler»responseGenerator
WARRequest»isXmlHttpRequest	WARRequestHandler»serverHostname
WARRequest»method	WARRequestHandler»serverPath
WARRequest»postFields	WARRequestHandler»serverPort
WARRequest»queryFields	WARRequestHandler»serverProtocol
WARRequest»setBody:	WARRequestHandler»setFilter:
WARRequest»setCookies:	WARRequestHandler»setParent:

WARequestHandler»url	WAServerAdaptor»requestHandler
WResponse class»messageForStatus:	WAServerAdaptor»responseFor:
WResponse class»statusFound	WAServerAdaptor»start
WResponse»contentType:	WAServerManager class»default
WResponse»contentType	WAServerManager»adaptors
WResponse»cookies	WAServerManager»canStart:
WResponse»destroy	WAServerManager»register:
WResponse»found	WAServerManager»start:
WResponse»headerAt:ifAbsent:	WASession»actionField
WResponse»headerAt:put:	WASession»actionUrlForContinuation:
WResponse»headers	WASession»actionUrlForKey:
WResponse»initializeOn:	WASession»application
WResponse»initialize	WASession»clearJumpTo
WResponse»location:	WASession»createCache
WResponse»redirectTo:	WASession»handleFiltered:
WResponse»status:message:	WASession»initializeFilters
WResponse»status:	WASession»initialize
WResponse»status	WASession»isSession
WResponseGenerator class»on:	WASession»presenter
WResponseGenerator»expiredRegistryKey	WASession»properties
WResponseGenerator»initializeOn:	WASession»start
WResponseGenerator»requestContext	WASession»unknownRequest
WResponseGenerator»request	WASession»updateRoot:
WResponseGenerator»respond	WASession»updateStates:
WResponseGenerator»response	WASession»updateUrl:
WRoot class»context:	WASession»url
WRoot»setContext:	WASessionContinuation»basicValue
WScriptGenerator»initialize	WASessionContinuation»captureAndInvoke
WScriptGenerator»loadScripts	WASessionContinuation»captureState
WScriptGenerator»writeLoadScriptsOn:	WASessionContinuation»redirectToContinuation:
WScriptGenerator»writeScriptTag:on:	WASessionContinuation»registerForUrl:
WAServerAdaptor class»default	WASessionContinuation»registerForUrl
WAServerAdaptor class»manager:	WASessionContinuation»request
WAServerAdaptor class»new	WASessionContinuation»respond:
WAServerAdaptor class»port:	WASessionContinuation»setStates:
WAServerAdaptor class»startOn:	WASessionContinuation»states
WAServerAdaptor»codec	WASessionContinuation»updateStates:
WAServerAdaptor»contextFor:	WASessionContinuation»updateUrl:
WAServerAdaptor»defaultPort	WASessionContinuation»value
WAServerAdaptor»defaultRequestHandler	WASessionContinuation»withUnregisteredHandlerDo:
WAServerAdaptor»handle:	WASnapshot»initialize
WAServerAdaptor»handlePadding:	WASnapshot»register:
WAServerAdaptor»handleRequest:	WASnapshot»reset
WAServerAdaptor»initializeWithManager:	WASnapshot»restore
WAServerAdaptor»initialize	WATagBrush»after
WAServerAdaptor»manager	WATagBrush»attributes
WAServerAdaptor»port:	WATagBrush»before
WAServerAdaptor»port	WATagBrush»close Tag
WAServerAdaptor»process:	WATagBrush»document
WAServerAdaptor»requestFor:	WATagBrush»isClosed

WATagBrush»openTag  
 WATagBrush»storeCallback:  
 WATagBrush»with:  
 WATagCanvas»space  
 WAUncescapedDocument»initializeWithStream:codec  
 WAUpdateRootVisitor class»root:  
 WAUpdateRootVisitor»initializeWithRoot:  
 WAUpdateRootVisitor»root  
 WAUpdateRootVisitor»visitPainter:  
 WAUpdateStatesVisitor class»snapshot:  
 WAUpdateStatesVisitor»initializeWithSnapshot:  
 WAUpdateStatesVisitor»snapshot  
 WAUpdateStatesVisitor»visitPresenter:  
 WAUpdateUrlVisitor class»url:  
 WAUpdateUrlVisitor»initializeWithUrl:  
 WAUpdateUrlVisitor»url  
 WAUpdateUrlVisitor»visitPainter:  
 WUrl class»absolute:  
 WUrl class»decodePercent:  
 WUrl»addAllToPath:  
 WUrl»addField:value:  
 WUrl»addField:  
 WUrl»addToPath:  
 WUrl»decode:  
 WUrl»decodedWith:  
 WUrl»encodeOn:  
 WUrl»encodePathOn:  
 WUrl»encodeQueryOn:  
 WUrl»encodeSchemeAndAuthorityOn:  
 WUrl»fragment  
 WUrl»initializeFromString:  
 WUrl»initialize  
 WUrl»isSeasideField:  
 WUrl»parsePath:  
 WUrl»parseQuery:  
 WUrl»password  
 WUrl»path:  
 WUrl»pathElementsIn:do:  
 WUrl»path  
 WUrl»postCopy  
 WUrl»printOn:  
 WUrl»queryFields:  
 WUrl»queryFields  
 WUrl»seasideUrl  
 WUrl»slash:  
 WUrl»subStringsIn:splitBy:do:  
 WUrl»user  
 WUrlEncoder class»on:codec:  
 WUrlEncoder»nextPutAll:  
 WUserConfiguration»addParent:  
 WUserConfiguration»canAddParent:  
 WUserConfiguration»expressionAt:ifAbsent:  
 WUserConfiguration»initialize  
 WUserConfiguration»localAttributeAt:ifAbsent:  
 WUserConfiguration»parents  
 WValueExpression»determineValueWithContext:configuration:  
 WValueExpression»value  
 WValueHolder class»with:  
 WValueHolder»contents:  
 WValueHolder»contents  
 WVisiblePresenterGuide»visitPresenter:  
 WVisitor»start:  
 WVisitor»visit:  
 WXmlDocument»closeTag:  
 WXmlDocument»destroy  
 WXmlDocument»initializeWithStream:codec:  
 WXmlDocument»openTag:attributes:closed:  
 WXmlDocument»openTag:attributes:  
 WXmlDocument»openTag:  
 WXmlDocument»print:  
 WXmlDocument»urlEncoder  
 WXmlDocument»xmlEncoder  
 WXmlEncoder»nextPutAll:  
 ZnSeasideServerAdaptorDelegate class»with:  
 ZnSeasideServerAdaptorDelegate»adaptor:  
 ZnSeasideServerAdaptorDelegate»adaptor  
 ZnSeasideServerAdaptorDelegate»handleRequest:  
 ZnZincServerAdaptor»basicStart  
 ZnZincServerAdaptor»configureDelegate  
 ZnZincServerAdaptor»configureServerForBinaryReading  
 ZnZincServerAdaptor»defaultCodec  
 ZnZincServerAdaptor»defaultDelegate  
 ZnZincServerAdaptor»defaultZnServer  
 ZnZincServerAdaptor»isStopped  
 ZnZincServerAdaptor»requestAddressFor:  
 ZnZincServerAdaptor»requestBodyFor:  
 ZnZincServerAdaptor»requestCookiesFor:  
 ZnZincServerAdaptor»requestFieldsFor:  
 ZnZincServerAdaptor»requestHeadersFor:  
 ZnZincServerAdaptor»requestMethodFor:  
 ZnZincServerAdaptor»requestUrlFor:  
 ZnZincServerAdaptor»requestVersionFor:  
 ZnZincServerAdaptor»responseFrom:  
 ZnZincServerAdaptor»server

## D Appendix: Method List of Seaside Counter Application with Empty Seed

List of methods extracted from the nurtured Web application when using an empty seed. This list includes all methods installed from Seaside framework, the Counter application and the base library of Pharo.

Array class»new:	BlockClosure»renderOn:
Array»isSelfEvaluating	BlockClosure»repeatWithGClf:
Array»printOn:	BlockClosure»repeat
Array»replaceFrom:to:with:startingAt:	BlockClosure»startpc
Array»shouldBePrintedAsLiteral	BlockClosure»value:value:value:
ArrayedCollection class»new:withAll:	BlockClosure»value:value:
ArrayedCollection class»new	BlockClosure»valueNoContextSwitch
ArrayedCollection class»with:with:with:	BlockClosure»valueWithArguments:
ArrayedCollection class»with:with:	BlockClosure»valueWithPossibleArguments:
ArrayedCollection class»with:	ByteArray»asByteArray
ArrayedCollection»mergeSortFrom:to:by:	ByteArray»replaceFrom:to:with:startingAt:
ArrayedCollection»size	ByteString class»compare:with:collated:
ArrayedCollection»sort:	ByteString class»findFirstInString:inSet:startingAt:
Association class»key:value:	ByteString class»indexOfAscii:inString:startingAt:
Association class»key:value:	ByteString class»stringHash:initialHash:
Association»expireWeakKey	ByteString class»translate:from:to:table:
Association»expiredWeakKey	ByteString»asByteArray
Association»key:WeakKey	ByteString»at:put:
Association»key:value:WeakKey	ByteString»at:
Association»keyWeakKey	ByteString»beginsWith:
Association»value:WeakKey	ByteString»byteAt:put:
Association»value:	ByteString»byteSize
Association»valueWeakKey	ByteString»findSubstring:in:startingAt:matchTable:
Association»value	ByteString»findSubstringViaPrimitive:in:startingAt:matchTable:
BlockClosure»argumentCount	ByteString»isByteString
BlockClosure»asContextWithSender:	ByteString»isOctetString
BlockClosure»asContext	ByteString»replaceFrom:to:with:startingAt:
BlockClosure»assert	ByteSymbol class»stringHash:initialHash:
BlockClosure»cull:	ByteSymbol»at:
BlockClosure»ensure:	ByteSymbol»findSubstring:in:startingAt:matchTable:
BlockClosure»fixCallbackTemps	ByteSymbol»isByteString
BlockClosure»forkAt:named:	ByteSymbol»privateAt:put:
BlockClosure»forkAt:	ByteSymbol»species
BlockClosure»ifCurtailed:	ByteSymbol»string:
BlockClosure»ifError:	CNGBTextConverter class»encodingNames
BlockClosure»isClosure	CP1250TextConverter class»encodingNames
BlockClosure»newProcess	CP1253TextConverter class»encodingNames
BlockClosure»numArgs	ChangesLog class»default
BlockClosure»numCopiedValues	ChangesLog»recordStartupStamp
BlockClosure»on:do:	Character class»codePoint:
BlockClosure»on:fork:	Character class»cr
BlockClosure»outerContext	Character class»lf



Character class»space	CompiledMethod»numTemps
Character class»value:	CompiledMethod»objectAt:
Character»=	CompiledMethod»primitive
Character»asCharacter	CompoundTextConverter class»encodingNames
Character»asInteger	ContextPart class»contextEnsure:
Character»asSymbol	ContextPart class»contextOn:do:
Character»asUppercase	ContextPart class»newForMethod:
Character»asciiValue	ContextPart class»theReturnMethod
Character»charCode	ContextPart»activateMethod:withArgs:receiver:class:
Character»characterSet	ContextPart»activateReturn:value:
Character»codePoint	ContextPart»at:put:
Character»digitValue	ContextPart»at:
Character»greaseInteger	ContextPart»bottomContext
Character»isAlphaNumeric	ContextPart»cut:
Character»isCharacter	ContextPart»doPop
Character»isDigit	ContextPart»exceptionClass
Character»isLetter	ContextPart»exceptionHandlerBlock
Character»isOctetCharacter	ContextPart»exceptionHandlerIsActive:
Character»isSeparator	ContextPart»exceptionHandlerIsActive
Character»isVowel	ContextPart»findContextSuchThat:
Character»leadingChar	ContextPart»findNextHandlerContextStarting
Character»to:	ContextPart»findNextUnwindContextUpTo:
Collection class»withAll:	ContextPart»handleSignal:
Collection»addAll:	ContextPart»insertSender:
Collection»allSatisfy:	ContextPart»isDead
Collection»anySatisfy:	ContextPart»jump
Collection»asArray	ContextPart»methodReturnTop
Collection»detect:ifNone:	ContextPart»nextHandlerContext
Collection»emptyCheck	ContextPart»pop
Collection»inject:into:	ContextPart»privSender:
Collection»isCollection	ContextPart»push:
Collection»isEmptyOrNil	ContextPart»pushTemporaryVariable:
Collection»isEmpty	ContextPart»releaseTo:
Collection»noneSatisfy:	ContextPart»resume:through:
Collection»notEmpty	ContextPart»resume:
Collection»printElementsOn:	ContextPart»return:from:
Collection»printNameOn:	ContextPart»return:through:
Collection»printOn:	ContextPart»return:
Collection»removeAll:	ContextPart»runUntilErrorOrReturnFrom:
Collection»removeAllFoundIn:	ContextPart»sender
Collection»sorted:	ContextPart»singleRelease
Collection»sorted	ContextPart»stackp:
CommandLineUIManager class»replacing:	ContextPart»stepToCallee
CommandLineUIManager»initialize	ContextPart»step
CommandLineUIManager»replacing:	ContextPart»terminateTo:
CompiledMethod»frameSize	Date class»fromSeconds:
CompiledMethod»header	Date class»fromString:
CompiledMethod»initialPC	Date class»readFrom:
CompiledMethod»isPrimitive	Date class»starting:
CompiledMethod»numLiterals	Date class»year:month:day:

Date»dayMonthYearDo:	DateAndTime»second
Date»monthIndex	DateAndTime»setJdn:seconds:nano:offset:
Date»printOn:format:	DateAndTime»ticks:offset:DateAndTime»dayOfWeek
Date»printOn:	DateAndTime»ticks
DateAndTime class»clock	DateAndTime»year
DateAndTime class»fromSeconds:offset:	Delay class»forMilliseconds:
DateAndTime class»fromSeconds:	Delay class»handleTimerEvent
DateAndTime class»fromString:	Delay class»initialize
DateAndTime class»initializeOffsets	Delay class»primSignal:atMilliseconds:
DateAndTime class»initialize	Delay class»restoreResumptionTimes
DateAndTime class»localOffset	Delay class»runTimerEventLoop
DateAndTime class»localTimeZone	Delay class»saveResumptionTimes
DateAndTime class»milliSecondsSinceMidnight	Delay class»scheduleDelay:
DateAndTime class»millisecondClockValue	Delay class»shutDown
DateAndTime class»now	Delay class»startTimerEventLoop
DateAndTime class»readFrom:	Delay class»startUp
DateAndTime class»readOptionalSeparatorFrom:	Delay class»stopTimerEventLoop
DateAndTime class»readTimezoneOffsetFrom:	Delay class»unscheduleDelay:
DateAndTime class»readTwoDigitIntegerFrom:	Delay»beingWaitedOn:
DateAndTime class»startUp:	Delay»beingWaitedOn
DateAndTime class»todayAtMilliSeconds:	Delay»delayDuration
DateAndTime class»waitForOffsets	Delay»resumptionTime:
DateAndTime class»year:month:day:hour:minute:offset:	Delay»resumptionTime
DateAndTime class»year:month:day:hour:minute:seconds:offset:	Delay»seconds:offset:
DateAndTime class»year:month:day:hour:minute:seconds:offset:	Delay»signalDelay:forSemaphore:
DateAndTime class»year:month:day:hour:minute:seconds:offset:	Delay»signalWaitingProcess
DateAndTime class»year:month:day:hour:minute:	Delay»unschedule
DateAndTime class»year:month:day:offset:	Delay»wait
DateAndTime class»year:month:day:	DelayWaitTimeout»isExpired
DateAndTime»<	DelayWaitTimeout»setDelay:forSemaphore:
DateAndTime»asDateAndTime	DelayWaitTimeout»signalWaitingProcess
DateAndTime»asTimeStamp	DelayWaitTimeout»wait
DateAndTime»asUTC	Dictionary»addAll:GRSmall
DateAndTime»dayMonthYearDo:	Dictionary»associationsDo:
DateAndTime»dayOfMonth	Dictionary»at:ifAbsent:GRSmall
DateAndTime»dayOfWeekAbbreviation	Dictionary»at:ifAbsent:Small
DateAndTime»dayOfWeekName	Dictionary»at:ifAbsent:
DateAndTime»hour24	Dictionary»at:ifAbsentPut:GRSmall
DateAndTime»hour	Dictionary»at:ifAbsentPut:
DateAndTime»julianDayNumber	Dictionary»at:ifPresent:GRSmall
DateAndTime»julianDayOffset	Dictionary»at:ifPresent:
DateAndTime»localSeconds	Dictionary»at:put:GRSmall
DateAndTime»minute	Dictionary»at:put:Small
DateAndTime»monthAbbreviation	Dictionary»at:put:
DateAndTime»monthName	Dictionary»at:
DateAndTime»month	Dictionary»do:
DateAndTime»normalize:ticks:base:	Dictionary»findIndexFor:GRSmall
DateAndTime»offset:DateAndTime»midnight	Dictionary»findIndexForKey:Small
DateAndTime»offset	Dictionary»fixCollisionsFrom:
DateAndTime»printHMSON:	Dictionary»includesKey:GRSmall

Dictionary»includesKey:	Exception class»,
Dictionary»initialize:GRSmall	Exception class»handles:
Dictionary»initializeSmall	Exception class»signal:
Dictionary»isEmptyGRSmall	Exception class»signal
Dictionary»keyAtValue:ifAbsent:	Exception»description
Dictionary»keysAndValuesDo:SmallDictionary	Exception»isResumable
class»new	Exception»messageText:
Dictionary»keysAndValuesDo:	Exception»messageText
Dictionary»keysAndValuesDo:	Exception»printOn:
Dictionary»keysDo:GRSmall	Exception»privHandlerContext:
Dictionary»noCheckAdd:	Exception»receiver
Dictionary»postCopyGRSmall	Exception»resume:
Dictionary»privateAt:put:GRSmall	Exception»resumeUnchecked:
Dictionary»privateAt:put:Small	Exception»signal:
Dictionary»rehash	Exception»signalerContext
Dictionary»removeKey:ifAbsent:	Exception»signal
Dictionary»scanFor:	ExceptionSet»,
Dictionary»seasideRequestFieldsGRSmall	ExceptionSet»add:
Dictionary»sizeGRSmall	ExceptionSet»handles:
Dictionary»valuesDo:	ExceptionSet»initialize
DiskStore class»checkVMVersion	ExtendedNumberParser»allowPlusSign
DiskStore class»initialize	ExtendedNumberParser»nextNumber
DiskStore class»reset	ExternalSemaphoreTable class»clearExternalObjects
DiskStore class»shutDown:	ExternalSemaphoreTable class»collectionBasedOn:withRoomFor:
DiskStore class»startUp:	ExternalSemaphoreTable class»freedSlotsIn:ratherThanIncreaseSizeTo:
DiskStore class»useFilePlugin	ExternalSemaphoreTable class»initialize
Duration class»days:hours:minutes:seconds:nanoSeconds:	ExternalSemaphoreTable class»registerExternalObject:
Duration class»days:hours:minutes:seconds:	ExternalSemaphoreTable class»safelyRegisterExternalObject:
Duration class»days:	ExternalSemaphoreTable class»safelyUnregisterExternalObject:
Duration class»nanoSeconds:	ExternalSemaphoreTable class»slotFor:in:
Duration class»seconds:nanoSeconds:	ExternalSemaphoreTable class»unprotectedExternalObjects:
Duration class»seconds:	ExternalSemaphoreTable class»unprotectedExternalObjects
Duration»+	ExternalSemaphoreTable class»unregisterExternalObject:
Duration»-	False»not
Duration»asDuration	False»
Duration»asMilliSeconds	FilePath class»pathName:isEncoded:
Duration»asNanoSeconds	FilePath»asSqueakPathName
Duration»asSeconds	FilePath»pathName:isEncoded:
Duration»days	FilePath»pathName
Duration»isZero	FileStream class»flushAndVoidStdioFiles
Duration»negated	FileStream class»initialize
Duration»seconds:nanoSeconds:	FileStream class»shutDown:
Duration»ticks	FileStream class»startUp:
DynamicVariable class»value:during:	FileStream class»stdioHandles
DynamicVariable»value:during:	FileStream class»voidStdioFiles
EUCJPTTextConverter class»encodingNames	Float class»precision
EUCKRTextConverter class»encodingNames	Float»adapToInteger:andSend:
EncodedCharSet class»charsetAt:	Float»asFloat
EventManager class»actionMaps	Float»isInfinite
EventManagerclass»flushEvents	Float»timesTwoPower:

Float»truncated	GRSmallDictionary class»withAll:
Fraction class»numerator:denominator:	HashTableSizes class»atLeast:
Fraction»>=	HashTableSizes class»sizes
Fraction»reduced	HashedCollection class»new:
Fraction»setNumerator:denominator:	HashedCollection class»new
Fraction»truncated	HashedCollection class»sizeFor:
GRCodecStream class»on:	HashedCollection»array
GRCodecStream»atEnd	HashedCollection»atNewIndex:put:
GRCodecStream»initializeOn:	HashedCollection»findElementOrNil:
GRNullCodec class»codecName	HashedCollection»fullCheck
GRNullCodec class»supportsEncoding:	HashedCollection»grow
GRNullCodec»encoderFor:	HashedCollection»initialize:
GRNullCodec»url	HashedCollection»size
GRNullCodecStream»nextPutAll:	Heap class»withAll:sortBy:
GRObject class»new	Heap»add:
GRObject»initialize	Heap»do:
GROrderedMultiMap»allAt:	Heap»downHeapSingle:
GROrderedMultiMap»at:add:	Heap»growHeap»reSort
GRPharoConverterCodecStream class»on:converter:	Heap»growSize
GRPharoConverterCodecStream»contents	Heap»growTo:
GRPharoConverterCodecStream»initializeOn:converter:	Heap»isEmpty
GRPharoConverterCodecStream»size	Heap»privateRemoveAt:
GRPharoGenericCodec class»supportedEncodingNames:	Heap»remove;ifAbsent:
GRPharoGenericCodec class»supportsEncoding:	Heap»removeFirst
GRPharoLatin1Codec class»supportedEncodingNames:	Heap»setCollection:tally:
GRPharoLatin1Codec class»supportsEncoding:	Heap»size
GRPharoPlatform»addToShutDownList:	Heap»sortBy:
GRPharoPlatform»addToStartUpList:	Heap»sorts:before:
GRPharoPlatform»includesUnsafeUrlCharacter:	Heap»upHeap:
GRPharoPlatform»includesUnsafeXmlCharacter:	Heap»updateObjectIndex:
GRPharoPlatform»semaphoreClass	ISO885915TextConverter class»encodingNames
GRPharoRandomProvider class»initialize	ISO88592TextConverter class»encodingNames
GRPharoRandomProvider class»nextInt:	IdentitySet»scanFor:
GRPharoRandomProvider class»randomClass	InstructionStream»interpretExtension:in:for:
GRPharoRandomProvider class»startUp	InstructionStream»interpretNextInstructionFor:
GRPharoUtf8Codec class»basicForEncoding:	Integer class»readFrom:base:
GRPharoUtf8Codec class»supportsEncoding:	Integer class»readFrom:
GRPharoUtf8Codec»decode:	Integer»*
GRPharoUtf8Codec»decoderFor:	Integer»+
GRPharoUtf8Codec»encoderFor:	Integer»<
GRPharoUtf8Codec»name	Integer»=
GRPharoUtf8Codec»url	Integer»>>
GRPharoUtf8CodecStream»encodeFast:	Integer»asCharacter
GRPharoUtf8CodecStream»next:	Integer»asInteger
GRPharoUtf8CodecStream»nextPut:	Integer»copyto:
GRPharoUtf8CodecStream»nextPutAll:	Integer»denominator
GRPlatform class»current	Integer»digitCompare:
GRPlatform»reducedConflictDictionary	Integer»digitDiv:neg:
GRSmallDictionary class»new:	Integer»digitMultiply:neg:
GRSmallDictionary class»new	Integer»digitSubtract:

Integer»floor	MethodContext»blockReturnTop
Integer»isFraction	MethodContext»closure
Integer»isInteger	MethodContext»contextTag
Integer»noMask:	MethodContext»methodReturnContext
Integer»normalize	MethodContext»method
Integer»numerator	MethodContext»privRefresh
Integer»printOn:	MethodContext»receiver
Integer»printStringBase:length:padded:	MethodContext»setSender:receiver:method:arguments:
Integer»printStringLength:padded:	MethodContext»setSender:receiver:method:closure:startpc:
Integer»quo:	MethodContext»tempAt:put:
Integer»rounded	MethodContext»tempAt:
Integer»timesRepeat:	MethodDictionary»at:ifAbsent:
Integer»truncated	MethodDictionary»includesKey:
Interval class»from:to:by:	MethodDictionary»scanFor:
Interval class»new	Month class»daysInMonth:forYear:
Interval»collect:	Month class»indexOfMonth:
Interval»setFrom:to:by:	Month class»nameOfMonth:
Interval»size	Mutex»critical:
Interval»species	Mutex»initialize
KOI8RTextConverter class»encodingNames	NetNameResolver class»initializeNetwork
LanguageEnvironment class»defaultFileNameConverter	NetNameResolver class»initialize
LargeNegativeInteger»negative	NetNameResolver class»primNameResolverStatus
LargeNegativeInteger»normalize	NetNameResolver class»resolverStatus
LargePositiveInteger>*	Number class»one
LargePositiveInteger»+	Number class»readFrom:
LargePositiveInteger»-	Number»//
LargePositiveInteger»//	Number»%
LargePositiveInteger»<	Number»\
LargePositiveInteger»asFloat	Number»abs
LargePositiveInteger»digitAt:	Number»asDuration
LargePositiveInteger»digitLength	Number»floor
LargePositiveInteger»highBitOfMagnitude	Number»fractionPart
LargePositiveInteger»negated	Number»integerPart
LargePositiveInteger»negative	Number»isNumber
LargePositiveInteger»normalize	Number»isZero
LargePositiveInteger»quo:	Number»negated
LimitedWriteStream»nextPut:	Number»negative
LimitedWriteStream»nextPutAll:	Number»quo:
LimitedWriteStream»setLimit:limitBlock:	Number»raisedToInteger:
LookupKey class»key:	Number»rem:
LookupKey»key:	Number»strictlyPositive
LookupKey»key	Number»to:
MacOSXPlatform class»isActivePlatform	NumberParser class»on:
MacRomanTextConverter class»encodingNames	NumberParser»nextElementaryLargeIntegerBase:
Magnitude»>	NumberParser»nextIntegerBase:
Magnitude»between:and:	NumberParser»nextUnsignedIntegerBase:
Magnitude»max:	NumberParser»nextUnsignedIntegerOrNilBase:
Magnitude»min:	NumberParser»on:
MethodContext class»sender:receiver:method:arguments:	NumberParser»peekSignIsMinus
MethodContext»aboutToReturn:through:	NumberParser»readExponent

OSPlatform class»determineActivePlatformStartingAt	Object»perform.withArguments.inSuperclass:
OSPlatform class»initialize	Object»perform.withArguments:
OSPlatform class»isMacOS	Object»perform:
OSPlatform class»platformName	Object»postCopy
OSPlatform class»shutDown:	Object»printOn:
OSPlatform class»startUp:ISO88597	Object»printStringLimitedTo:
OSPlatform class»version	Object»printString
OSPlatform»shutDown:	Object»readFromString:
OSPlatform»startUp:	Object»renderOn:
Object class»flushDependents	Object»respondsTo:
Object class»flushEvents	Object»restoreFromSnapshot:
Object class»initializeDependentsFields	Object»shallowCopy
Object class»initialize	Object»shouldBePrintedAsLiteral
Object class»newFrom:	Object»snapshotCopy
Object»=	Object»species
Object»actAsExecutor	Object»split.do:
Object»as:	Object»split:
Object»asSetElement	Object»value
Object»asString	Object»yourself
Object»assert:	Object»~=
Object»at.put:	OrderedCollection class»arrayType
Object»at:	OrderedCollection class»new:
Object»basicAt.put:	OrderedCollection class»new
Object»basicAt:	OrderedCollection»add:
Object»basicSize	OrderedCollection»addAll:
Object»breakDependents	OrderedCollection»addAllLast:
Object»class	OrderedCollection»addFirst:
Object»copyFrom:	OrderedCollection»addLast:
Object»copySameFrom:	OrderedCollection»asArray
Object»copy	OrderedCollection»at:
Object»enclosedSetElement	OrderedCollection»collect:
Object»encodeOn:	OrderedCollection»copyEmpty
Object»executor	OrderedCollection»do:
Object»greaseString	OrderedCollection»ensureBoundsFrom:to:
Object»hash	OrderedCollection»growAtLast
Object»instVarAt.put:	OrderedCollection»makeRoomAtFirst
Object»instVarAt:	OrderedCollection»makeRoomAtLast
Object»instVarNamed.put:	OrderedCollection»postCopy
Object»isArray	OrderedCollection»remove.ifAbsent:
Object»isCharacter	OrderedCollection»removeFirst
Object»isInteger	OrderedCollection»removeIndex:
Object»isKindOf:	OrderedCollection»resetTo:
Object»isLiteral	OrderedCollection»reset
Object»isMemberOf:	OrderedCollection»reverseDo:
Object»isSelfEvaluating	OrderedCollection»select:
Object»isString	OrderedCollection»setCollection:
Object»isSymbol	OrderedCollection»size
Object»myDependents:	PositionableStream class»on:
Object»notNil	PositionableStream»atEnd
Object»perform.with:	PositionableStream»isBinary

PositionableStream»isEmpty	ProtoObject»identityHash
PositionableStream»on:	ProtoObject»initialize
PositionableStream»originalContents	ProtoObject»instVarsInclude:
PositionableStream»peekFor:	ProtoObject»isNil
PositionableStream»peek	ProtoObject»pointsTo:
PositionableStream»position:	ProtoObject»~~
PositionableStream»position	Random»initialize
PositionableStream»reset	Random»nextInt:
PositionableStream»skip:	Random»nextValue
PositionableStream»skipSeparators	Random»next
PositionableStream»skipTo:	ReadStream class»on:from:to:
Process class»forContext:priority:	ReadStream»next
Process»activateReturn:value:	ReadStream»on:from:to:
Process»calleeOf:	ReadStream»upTo:
Process»complete:	ReadStream»upToEnd
Process»isActiveProcess	Semaphore class»forMutualExclusion
Process»name:	Semaphore class»new
Process»popTo:	Semaphore»critical:ifError:
Process»primitiveResume	Semaphore»critical:
Process»priority:	Semaphore»initSignals
Process»priority	Semaphore»signal
Process»psValueAt:put:	Semaphore»waitTimeoutMsecs:
Process»psValueAt:	Semaphore»wait
Process»resume	SequenceableCollection class»new:streamContents:
Process»return:value:	SequenceableCollection class»ofSize:
Process»suspendedContext:	SequenceableCollection class»streamContents:limitedTo:
Process»suspendingList	SequenceableCollection class»streamContents:
Process»suspend	SequenceableCollection class»streamSpecies
Process»terminate	SequenceableCollection»,
ProcessLocalVariable class»value:	SequenceableCollection»allButFirst:
ProcessLocalVariable»value:	SequenceableCollection»at:ifAbsent:
ProcessSpecificVariable class»soleInstance	SequenceableCollection»atAllPut:
ProcessSpecificVariable class»value	SequenceableCollection»copyAfter:
ProcessSpecificVariable»default	SequenceableCollection»copyFrom:to:
ProcessSpecificVariable»value	SequenceableCollection»copyReplaceFrom:to:with:
ProcessorScheduler class»idleProcess	SequenceableCollection»copyUpTo:
ProcessorScheduler class»initialize	SequenceableCollection»do:separatedBy:
ProcessorScheduler class»relinquishProcessorForMsecs»relinquish:	SequenceableCollection»do:
ProcessorScheduler class»startUp	SequenceableCollection»doWithIndex:
ProcessorScheduler»activePriority	SequenceableCollection»first:
ProcessorScheduler»activeProcess	SequenceableCollection»first
ProcessorScheduler»highIOPriority	SequenceableCollection»from:to:put:
ProcessorScheduler»highestPriority	SequenceableCollection»grownBy:
ProcessorScheduler»lowIOPriority	SequenceableCollection»includes:
ProcessorScheduler»lowestPriority	SequenceableCollection»indexOf:ifAbsent:
ProcessorScheduler»terminateActive	SequenceableCollection»indexOf:startingAt:ifAbsent:
ProcessorScheduler»timingPriority	SequenceableCollection»indexOf:
ProcessorScheduler»userInterruptPriority	SequenceableCollection»indexOfSubCollection:startingAt:
ProtoObject»basicIdentityHash	SequenceableCollection»keysAndValuesDo:
ProtoObject»flag:	SequenceableCollection»last

SequenceableCollection»readStream	SmalltalkImage»logStartupErrorDuring:into:tryDebugger:
SequenceableCollection»replaceFrom:to:with:	SmalltalkImage»lowSpaceThreshold
SequenceableCollection»reverseDo:	SmalltalkImage»lowSpaceWatcher
SequenceableCollection»second	SmalltalkImage»newSessionObject
SequenceableCollection»select:	SmalltalkImage»primImagePath
SequenceableCollection»split:indicesDo:	SmalltalkImage»primSignalAtBytesLeft:
SequenceableCollection»splitOn:	SmalltalkImage»primitiveGetSpecialObjectsArray
SequenceableCollection»swap:with:	SmalltalkImage»processShutDownList:
SequenceableCollection»withIndexDo:	SmalltalkImage»processStartupList:
SequenceableCollection»writeStream	SmalltalkImage»recordStartupStamp
Set class»new	SmalltalkImage»registerExternalObject:
Set»add:	SmalltalkImage»send:toClassesNamedIn:with:
Set»do:	SmalltalkImage»shutDownImage:
Set»fixCollisionsFrom:	SmalltalkImage»specialObjectsArray
Set»grow	SmalltalkImage»startupImage:snapshotWorked:
Set»includes:	SmalltalkImage»unregisterExternalObject:
Set»noCheckAdd:	SmalltalkImage»vm
Set»remove:ifAbsent:	SmalltalkImage»wordSize
Set»scanFor:	Socket class»acceptFrom:
SmallInteger class»maxValCP1252	Socket class»initializeNetwork
SmallInteger»*	Socket class»initialize
SmallInteger»/	Socket class»newTCP
SmallInteger»asFloat	Socket class»register:
SmallInteger»decimalDigitLength	Socket class»registry
SmallInteger»gcd:	Socket class»standardTimeout
SmallInteger»hashMultiply	Socket class»unregister:
SmallInteger»hash	Socket»acceptFrom:
SmallInteger»highBitOfPositiveReceiver	Socket»accept
SmallInteger»highBit	Socket»closeAndDestroy:
SmallInteger»identityHash	Socket»closeAndDestroy
SmallInteger»isLarge	Socket»close
SmallInteger»numberOfDigitsInBase:	Socket»dataAvailable
SmallInteger»printOn:base:length:padded:	Socket»destroy
SmallInteger»printOn:base:	Socket»initialize:
SmallInteger»printString	Socket»isConnected
SmallInteger»quo:	Socket»isOtherEndClosed
SmalltalkImage class»initializeForTornado	Socket»isValid
SmalltalkImage»addDeferredStartupAction:	Socket»listenOn:backlogSize:
SmalltalkImage»addToShutDownList:	Socket»primAcceptFrom:receiveBufferSize:sendBufSize:semalIndex:readSemalIndex:writeSemalIndex:
SmalltalkImage»addToStartupList:	Socket»primSocket:receiveDataInto:startingAt:count:
SmalltalkImage»at:ifAbsent:	Socket»primSocket:sendData:startIndex:count:
SmalltalkImage»at:	Socket»primSocket:setOption:value:
SmalltalkImage»clearExternalObjects	Socket»primSocketConnectionStatus:
SmalltalkImage»executeDeferredStartupActions:	Socket»primSocketDestroy:
SmalltalkImage»garbageCollectMost	Socket»primSocketReceiveDataAvailable:
SmalltalkImage»imagePath	Socket»primSocketRemoteAddress:
SmalltalkImage»includesKey:	Socket»primSocketSendDone:
SmalltalkImage»installLowSpaceWatcher	Socket»readSemaphore
SmalltalkImage»isHeadless	Socket»register
SmalltalkImage»isInteractive	Socket»remoteAddress



Socket»sendSomeData:startIndex:count:for:	String»isString
Socket»sendSomeData:startIndex:count:	String»isWideString
Socket»setOption:value:	String»match:
Socket»socketHandle	String»putOn:
Socket»unregister	String»renderOn:
Socket»waitForAcceptFor:	String»sameAs:
Socket»waitForConnectionFor:ifTimedOut:	String»seasideMimeType
Socket»waitForDataFor:ifClosed:ifTimedOut:	String»startingAt:match:startingAt:
Socket»waitForDataFor:	String»subStrings:
Socket»waitForDisconnectionFor:	String»translateFrom:to:table:
Socket»waitForSendDoneFor:	String»translateToLowercase
SparseLargeTable»at:	String»translateToUppercase
SparseLargeTable»noCheckAt:	String»translateWith:
SparseLargeTable»pvtCheckIndex:	String»trimBoth:
SparseLargeTable»size	String»trimBoth
SqNumberParser»allowPlusSign	String»trimLeft:right:
SqNumberParser»makeIntegerOrScaledInteger	Symbol class»initializeForTornadoWATagCanvas»space
SqNumberParser»readScale	Symbol class»intern:
Stream»basicNext	Symbol class»internCharacter:
Stream»nextPutAll:	Symbol class»lookup:
Stream»print:	Symbol class»shutDown:
String class»crlf	Symbol»=
String class»empty	Symbol»asString
String class»new:	TextConverter class»allEncodingNames
String:startingAt:	TextConverter class»encodingNames
String»=	TextConverter class»encodingNames
String»asDateAndTime	TextConverter class»encodingNames
String»asDate	TextConverter class»encodingNames
String»asLowercase	TextConverter class»encodingNames
String»asNumber	TextConverter class»encodingNames
String»asString	TextConverter class»latin1Encodings
String»asSymbol	TextConverter class»latin1MapGRCodec
String»asUppercase	class»forEncoding:
String»asZnMimeType	TextConverter»initialize
String»asZnUrl	TextConverter»nextFromStream:UTF8
String»compare:caseSensitive:	Time class»dateAndTimeFromSeconds:
String»compare:with:collated:	Time class»dateAndTimeNow
String»convertFromWithConverter:	Time class»fromSeconds:
String»encodeOn:	Time class»hour:minute:second:nanoSecond:
String»findString:startingAt:caseSensitive:	Time class»millisecondClockValue
String»findString:	Time class»milliseconds:since:
String»find	Time class»millisecondsSince:
String»greaseInteger	Time class»primSecondsClock
String»hash	Time class»readFrom:
String»includesSubstring:	Time class»seconds:nanoSeconds:
String»indexOf:startingAt:ifAbsent:	Time class»secondsWhenClockTicks
String»indexOf:startingAt:	Time class»totalSeconds
String»indexOf:	Time»hour24
String»indexOfSubCollection:startingAt:ifAbsent:	Time»hour
String»indexOfSubCollection:	Time»minute

Time»nanoSecond	VirtualMachine class»version
Time»print24:showSeconds:on:	VirtualMachine class»wordSize
Time»printOn:	WAAccessIntervalReapingStrategy»defaultConfiguration
Time»seconds	WAAccessIntervalReapingStrategy»initialize
Time»second	WAAccessIntervalReapingStrategy»interval
Time»ticks:	WAAccessIntervalReapingStrategy»reap
TimeZone»offset	WAAccessIntervalReapingStrategy»stored:key:
Timespan class»starting:duration:	WAActionCallback»block:
Timespan»<	WAActionCallback»evaluateWithArgument:
Timespan»dayOfMonth	WAActionCallback»isEnabledFor:
Timespan»duration:	WAActionCallback»signalRenderNotification
Timespan»month	WAActionPhaseContinuation»continue
Timespan»start:	WAActionPhaseContinuation»handleRequest
Timespan»start	WAActionPhaseContinuation»renderContext:
Timespan»year	WAActionPhaseContinuation»renderContext
True»ifFalse:	WAActionPhaseContinuation»runCallbacks
True»not	WAActionPhaseContinuation»shouldRedirect
True»	WAdmin class»defaultServerManager
UIManager class»basicDefault:	WAdmin class»serverAdaptors
UIManager class»default:WAUnescapedDocument»initializeWithGreatUnicode:	WAnchorTag»tag
UIManager class»default	WAnchorTag»tag
UIManager»activate	WAnchorTag»url
UIManager»beDefault	WAnchorTag»with:
UIManager»boot:during:	WAApplication»contentType
UIManager»deactivate	WAApplication»doesHandlerSupportCookies:
UIManager»onSnapshot:	WAApplication»handleDefault:
UTF16TextConverter class»encodingNames	WAApplication»handleFiltered:
UTF8DecomposedTextConverter class»encodingNames	WAApplication»isApplication
UUIDGenerator class»initialize	WAApplication»isImplemented:
UUIDGenerator class»startUp	WAApplication»keyField
UndefinedObject»encodeOn:	WAApplication»libraries
UndefinedObject»isNil	WAApplication»mainClass
UndefinedObject»notNil	WAApplication»mimeType
UndefinedObject»seasideUrl	WAApplication»newSession
UndefinedObject»shallowCopy	WAApplication»resourceBaseUrl
Unicode class»isDigit:	WAApplication»sessionClass
Unicode class»isLetter:	WAApplicationConfiguration»parents
Unicode class»toUppercase:	WAAtributeSearchContext class»key:target:
VirtualMachine class»allocationsBetweenGC:	WAAtributeSearchContext»at:ifPresent:
VirtualMachine class»getSystemAttribute:	WAAtributeSearchContext»at:put:
VirtualMachine class»interpreterClass	WAAtributeSearchContext»attribute
VirtualMachine class»interpreterSourceDate	WAAtributeSearchContext»cachedValues
VirtualMachine class»interpreterSourceVersion	WAAtributeSearchContext»findAttributeAndSelectAncestorsOf:
VirtualMachine class»isPharoVM	WAAtributeSearchContext»initializeWithKey:
VirtualMachine class»isRunningCogit	WAAtributeSearchContext»isAttributeInheritedOn:
VirtualMachine class»maxExternalSemaphores	WAAtributeSearchContext»isAttributeLocalOn:
VirtualMachine class»parameterAt:put:	WAAtributeSearchContext»key
VirtualMachine class»parameterAt:	WABrush»initialize
VirtualMachine class»setGCPParameters	WABrush»parent
VirtualMachine class»tenuringThreshold:	WABrush»setParent:canvas:

WABrush»with:  
 WABufferedResponse class»on:  
 WABufferedResponse»contents  
 WABufferedResponse»destroy  
 WABufferedResponse»initializeOn:  
 WABufferedResponse»stream  
 WACache»at:ifAbsent:  
 WACache»expiryPolicy  
 WACache»initializeCollections  
 WACache»initializeMutex  
 WACache»initialize  
 WACache»keyAtValue:ifAbsent:  
 WACache»keyAtValue:  
 WACache»keySize  
 WACache»missStrategy  
 WACache»notifyRemoved:key:  
 WACache»notifyRetrieved:key:  
 WACache»notifyStored:key:  
 WACache»pluginsDo:  
 WACache»reapingStrategy  
 WACache»reap  
 WACache»removalAction  
 WACache»setExpiryPolicy:  
 WACache»setMissStrategy:  
 WACache»setReapingStrategy:  
 WACache»setRemovalAction:  
 WACache»store:  
 WACacheCapacityConfiguration»describeOn:  
 WACachePlugin»configuration  
 WACachePlugin»defaultConfiguration  
 WACachePlugin»initialize  
 WACachePlugin»removed:key:  
 WACachePlugin»retrieved:key:  
 WACachePlugin»setCache:  
 WACachePlugin»stored:key:  
 WACacheReapingStrategy»reap  
 WACallback class»on:  
 WACallback»convertKey:  
 WACallback»evaluateWithFieldValues:  
 WACallback»key  
 WACallback»setKey:callbacks:  
 WACallback»valueForField:  
 WACallbackRegistry»advanceKey  
 WACallbackRegistry»handle:  
 WACallbackRegistry»increaseKey  
 WACallbackRegistry»initialize  
 WACallbackRegistry»nextKey  
 WACallbackRegistry»store:  
 WACanvas»brush:  
 WACanvas»flush  
 WACanvas»nest:  
 WACanvas»render:  
 WACanvas»text:  
 WAComponent»accept:  
 WAComponent»acceptDecorated:  
 WAComponent»decoration  
 WAComponent»initialize  
 WAComponent»updateStates:  
 WAConfigurationDescription»add:to:  
 WAConfigurationDescription»addAttribute:  
 WAConfigurationDescription»attributes  
 WAConfigurationDescription»expressions  
 WAConfigurationDescription»initialize  
 WAConfigurationDescription»integer:  
 WAConfiguredRequestFilter»configuration  
 WACounter»count:  
 WACounter»decrease  
 WACounter»increase  
 WACounter»initialize  
 WACounter»renderContentOn:  
 WACounter»states  
 WADefaultScriptGenerator»close:on:  
 WADefaultScriptGenerator»open:on:  
 WADevelopmentConfiguration»parents  
 WADispatcher class»default  
 WADispatcher»handleFiltered:named:  
 WADispatcher»handleFiltered:  
 WADispatcher»handlerAt:ifAbsent:  
 WADispatcher»handlerAt:with:  
 WADispatcher»handlers  
 WADispatcher»nameOfHandler:  
 WADispatcher»urlFor:  
 WADocument class»on:codec:  
 WADocument class»on:  
 WADocument»closeWADocument»destroy  
 WADocument»initializeWithStream:codec:  
 WADocument»nextPut:  
 WADocument»nextPutAll:  
 WADocument»open:  
 WADynamicVariable class»use:during:  
 WADynamicVariable class»value  
 WAEncoder class»on:table:  
 WAEncoder class»on:  
 WAEncoder»initializeOn:table:  
 WAEncoder»nextPut:  
 WAErrorHandler class»exceptionSelector  
 WAExampleComponent»rendererClass  
 WAExceptionFilter»exceptionHandler  
 WAExceptionFilter»handleFiltered:  
 WAExceptionHandler class»context:

WAExceptionHandler class»exceptionSelector	WAKeyGenerator class»current
WAExceptionHandler class»handleExceptionsDuring	WAKeyGenerator»keyOfLength:
WAExceptionHandler class»handles:	WALastAccessExpiryPolicy»defaultConfiguration
WAExceptionHandler»handleExceptionsDuring:	WALastAccessExpiryPolicy»initialize
WAExceptionHandler»handles:	WALastAccessExpiryPolicy»isExpired:key:
WAExceptionHandler»initializeWithContext:	WALastAccessExpiryPolicy»retrieved:key:
WAHeaderFields»checkValue:	WALastAccessExpiryPolicy»stored:key:
WAHeaderFields»privateAt:put:	WALastAccessExpiryPolicy»timeout
WAHeadingTag»initialize	WALeastRecentlyUsedExpiryPolicy»defaultConfiguration
WAHeadingTag»level1	WALeastRecentlyUsedExpiryPolicy»initialize
WAHeadingTag»level	WALeastRecentlyUsedExpiryPolicy»isExpired:key:
WAHeadingTag»tag	WALeastRecentlyUsedExpiryPolicy»maximumAge
WAHtmlAttributes»encodeOn:	WALeastRecentlyUsedExpiryPolicy»removed:key:
WAHtmlAttributes»privateAt:put:	WALeastRecentlyUsedExpiryPolicy»retrieved:key:
WAHtmlCanvas»anchor	WALeastRecentlyUsedExpiryPolicy»stored:key:
WAHtmlCanvas»heading:	WAMergedRequestFields class»on:
WAHtmlCanvas»heading	WAMergedRequestFields»allAt:
WAHtmlCanvas»spaceEntity	WAMergedRequestFields»at:ifAbsent:
WAHtmlDocument»scriptGenerator:	WAMergedRequestFields»includesKey:
WAHtmlDocument»scriptGenerator	WAMergedRequestFields»initializeOn:
WAHtmlElement class»root:	WAMergedRequestFields»keysDo:
WAHtmlElement»attributeAt:put:	WAMetaElement»content:
WAHtmlElement»attributes	WAMetaElement»contentScriptType:
WAHtmlElement»encodeBeforeOn:	WAMetaElement»contentType:
WAHtmlElement»encodeOn:	WAMetaElement»encodeBeforeOn:
WAHtmlElement»initializeWithRoot:	WAMetaElement»responseHeaderName:
WAHtmlElement»isClosed	WAMetaElement»tag
WAHtmlRoot»add:	WAMimeType class»fromString:
WAHtmlRoot»beXhtml10Strict	WAMimeType class»main:sub:
WAHtmlRoot»bodyAttributes	WAMimeType class»text.Javascript
WAHtmlRoot»closeOn:	WAMimeType class»text.Plain
WAHtmlRoot»docType:	WAMimeType»charset:
WAHtmlRoot»htmlAttributes	WAMimeType»greaseString
WAHtmlRoot»initialize	WAMimeType»main:
WAHtmlRoot»meta	WAMimeType»main
WAHtmlRoot»openOn:	WAMimeType»parameters
WAHtmlRoot»title:	WAMimeType»sub:
WAHtmlRoot»writeElementsOn:	WAMimeType»sub
WAHtmlRoot»writeFootOn:	WAMutex»critical:
WAHtmlRoot»writeHeadOn:	WAMutex»initialize
WAHtmlRoot»writeScriptsOn:	WAMutualExclusionFilter»handleFiltered:
WAHtmlRoot»writeStylesOn:	WAMutualExclusionFilter»initialize
WAHttpVersion class»fromString:	WAMutualExclusionFilter»shouldTerminate:
WAHttpVersion class»major:minor:	WANotifyRemovalAction»removed:key:
WAHttpVersion class»readFrom:	WAObject»application
WAHttpVersion»initializeWithMajor:minor:	WAObject»requestContext
WAInitialRequestVisitor class»request:	WAObject»session
WAInitialRequestVisitor»initializeWithRequest:	WAPainter»renderWithContext:
WAInitialRequestVisitor»request	WAPainter»updateRoot:
WAInitialRequestVisitor»visitPresenter:	WAPainter»updateUrl:

WAPainterVisitor»visitComponent:	WARenderPhaseContinuation»processRendering:
WAPainterVisitor»visitDecorationsOfComponent:	WARenderVisitor class»context:
WAPainterVisitor»visitPainter:	WARenderVisitor»initializeWithContext:
WAPainterVisitor»visitPresenter:	WARenderVisitor»renderContext
WAPathConsumer class»path:	WARenderVisitor»visitPainter:
WAPathConsumer»atEnd	WARenderer class»context:
WAPathConsumer»initializeWith:	WARenderer»actionUrl
WAPathConsumer»next	WARenderer»callbacks
WAPresenter»childrenDo:	WARenderer»context
WAPresenter»children	WARenderer»document
WAPresenter»initialRequest:	WARenderer»flush
WAPresenter»script	WARenderer»initializeWithContext:
WAPresenter»style	WARenderer»render:
WAPresenter»updateRoot:	WARenderer»text:
WAPresenter»updateStates:	WARRequest class»method.uri.version:GRSmall
WAPresenterGuide class»client:	WARRequest»at.ifAbsent:
WAPresenterGuide»client	WARRequest»cookiesAt:
WAPresenterGuide»initializeWithClient:	WARRequest»cookies
WAPresenterGuide»visit:	WARRequest»destroy
WAPresenterGuide»visitPainter:	WARRequest»fields
WARegistryConfiguration»parents	WARRequest»headerAt.ifAbsent:
WARenderContext»actionBaseUrl:	WARRequest»headerAt:
WARenderContext»actionUrl:	WARRequest»initializeWithMethod.uri.version:
WARenderContext»actionUrl	WARRequest»isGet
WARenderContext»callbacks	WARRequest»isPrefetch
WARenderContext»defaultVisitor	WARRequest»isXmlHttpRequest
WARenderContext»destroy	WARRequest»method
WARenderContext»document:	WARRequest»postFields
WARenderContext»document	WARRequest»queryFields
WARenderContext»initialize	WARRequest»setBody:
WARenderContext»resourceUrl:	WARRequest»setCookies:
WARenderContext»visitor:	WARRequest»setHeaders:
WARenderContext»visitor	WARRequest»setPostFields:
WARenderLoopConfiguration»parents	WARRequest»setRemoteAddress:
WARenderLoopContinuation»createActionContinuation:	WARRequest»uri
WARenderLoopContinuation»createRenderContinuation:	WARRequest»url
WARenderLoopContinuation»presenter	WARRequestContext class»request.response.codec:
WARenderLoopContinuation»toPresenterSendRoot:	WARRequestContext»application
WARenderLoopContinuation»updateRoot:	WARRequestContext»charSet
WARenderLoopContinuation»updateStates:	WARRequestContext»codec
WARenderLoopContinuation»updateUrl:	WARRequestContext»consumer
WARenderLoopContinuation»withNotificationHandle:	WARRequestContext»destroy
WARenderLoopMain»createRoot	WARRequestContext»handlers
WARenderLoopMain»prepareRoot:	WARRequestContext»handler
WARenderLoopMain»rootClass	WARRequestContext»initializeWithRequest.response.codec:
WARenderLoopMain»rootDecorationClasses	WARRequestContext»newDocument
WARenderLoopMain»start	WARRequestContext»push.during:
WARenderPhaseContinuation»createHtmlRootWithContext:	WARRequestContext»request
WARenderPhaseContinuation»createRenderContext:	WARRequestContext»respond:
WARenderPhaseContinuation»handleRequest	WARRequestContext»respond

WRequestContext»responseGenerator	WResponse»status
WRequestContext»response	WResponseGenerator class»on:ShiftJIS
WRequestContext»session	WResponseGenerator»initializeOn:
WRequestFilter»handleFiltered:	WResponseGenerator»notFound
WRequestFilter»initialize	WResponseGenerator»requestContext
WRequestFilter»next	WResponseGenerator»request
WRequestFilter»setNext:	WResponseGenerator»respond
WRequestFilter»updateStates:	WResponseGenerator»response
WRequestHandler»addFilter:	WRoot class»context:
WRequestHandler»addFilterLast:	WRoot»setContext:
WRequestHandler»basicUrl	WScriptGenerator»initialize
WRequestHandler»configuration:	WScriptGenerator»loadScripts
WRequestHandler»configuration	WScriptGenerator»writeLoadScriptsOn:
WRequestHandler»defaultConfiguration	WScriptGenerator»writeScriptTag:on:
WRequestHandler»documentClass	WServerAdaptor class»defaultSmall
WRequestHandler»filters	WServerAdaptor class»manager:
WRequestHandler»filter	WServerAdaptor class»new
WRequestHandler»handle:	WServerAdaptor class»port:
WRequestHandler»initialize	WServerAdaptor class»startOn:
WRequestHandler»isApplication	WServerAdaptor»codec
WRequestHandler»isRoot	WServerAdaptor»contextFor:
WRequestHandler»parent	WServerAdaptor»defaultPort
WRequestHandler»preferenceAt:	WServerAdaptor»defaultRequestHandler
WRequestHandler»responseGenerator	WServerAdaptor»handle:
WRequestHandler»serverHostname	WServerAdaptor»handlePadding:
WRequestHandler»serverPath	WServerAdaptor»handleRequest:
WRequestHandler»serverPort	WServerAdaptor»initializeWithManager:
WRequestHandler»serverProtocol	WServerAdaptor»initialize
WRequestHandler»setFilter:	WServerAdaptor»manager
WRequestHandler»setParent:	WServerAdaptor»port:
WRequestHandler»url	WServerAdaptor»port
WResponse class»messageForStatus:	WServerAdaptor»process:
WResponse class»statusFound	WServerAdaptor»requestFor:
WResponse class»statusNotFound	WServerAdaptor»requestHandler
WResponse»contentType:	WServerAdaptor»responseFor:
WResponse»contentType	WServerAdaptor»start
WResponse»cookies	WServerManager class»default
WResponse»destroy	WServerManager class»initialize
WResponse»found	WServerManager class»shutDown
WResponse»headerAt:ifAbsent:	WServerManager class»startUp
WResponse»headerAt:put:	WServerManager»adaptors
WResponse»headers	WServerManager»canStart:
WResponse»initializeOn:	WServerManager»register:
WResponse»initialize	WServerManager»start:
WResponse»location:	WSession»actionField
WResponse»nextPutAll:	WSession»actionUrlForContinuation:
WResponse»notFound	WSession»actionUrlForKey:
WResponse»redirectTo:	WSession»application
WResponse»status:message:	WSession»clearJumpTo
WResponse»status:	WSession»createCache

WASession»handleFiltered:	WUpdateUrlVisitor»visitPainter:
WASession»initializeFilters	WUrl class»absolute:
WASession»initialize	WUrl class»decodePercent:
WASession»isSession	WUrl»addField:value:
WASession»presenter	WUrl»addField:
WASession»properties	WUrl»addToPath:
WASession»start	WUrl»decode:
WASession»updateRoot:	WUrl»decodedWith:
WASession»updateStates:	WUrl»encodeOn:
WASession»updateUrl:	WUrl»encodePathOn:
WASession»url	WUrl»encodeQueryOn:
WASessionContinuation»basicValue	WUrl»encodeSchemeAndAuthorityOn:
WASessionContinuation»captureAndInvoke	WUrl»fragment
WASessionContinuation»captureState	WUrl»initializeFromString:
WASessionContinuation»redirectToContinuation:	WUrl»initialize
WASessionContinuation»registerForUrl:	WUrl»parsePath:
WASessionContinuation»registerForUrl	WUrl»parseQuery:
WASessionContinuation»request	WUrl»password
WASessionContinuation»respond:	WUrl»path:
WASessionContinuation»setStates:	WUrl»pathElementsIn:do:
WASessionContinuation»states	WUrl»pathString
WASessionContinuation»updateStates:	WUrl»path
WASessionContinuation»updateUrl:	WUrl»postCopy
WASessionContinuation»value	WUrl»printOn:
WASessionContinuation»withUnregisteredHandlerDefault:	WUrl»queryFields:
WASlime class»initialize	WUrl»queryFields
WASlime class»startUp	WUrl»seasideUrl
WASnapshot»initialize	WUrl»slash:
WASnapshot»register:	WUrl»subStringsIn:splitBy:do:
WASnapshot»resetWASnapshot»restore	WUrl»user
WATagBrush»after	WUrlEncoder class»on:codec:
WATagBrush»attributes	WUrlEncoder»nextPutAll:
WATagBrush»before	WUserConfiguration»addParent:
WATagBrush»closeTag	WUserConfiguration»canAddParent:
WATagBrush»document	WUserConfiguration»expressionAt:ifAbsent:
WATagBrush»isClosed	WUserConfiguration»initialize
WATagBrush»openTag	WUserConfiguration»localAttributeAt:ifAbsent:
WATagBrush»storeCallback:	WUserConfiguration»parents
WATagBrush»with:	WValueExpression»determineValueWithContext:configuration:
WUpdateRootVisitor class»root:WeakKey	WValueExpression»value
WUpdateRootVisitor»initializeWithRoot:	WValueHolder class»with:
WUpdateRootVisitor»root	WValueHolder»contents:
WUpdateRootVisitor»visitPainter:	WValueHolder»contents
WUpdateStatesVisitor class»snapshot:	WVisiblePresenterGuide»visitPresenter:
WUpdateStatesVisitor»initializeWithSnapshot:	WVisitor»visit:WVisitor»start:
WUpdateStatesVisitor»snapshot	WXmlDocument»closeTag:
WUpdateStatesVisitor»visitPresenter:	WXmlDocument»destroy
WUpdateUrlVisitor class»url:	WXmlDocument»initializeWithStream:codec:
WUpdateUrlVisitor»initializeWithUrl:	WXmlDocument»openTag:attributes:closed:
WUpdateUrlVisitor»url	WXmlDocument»openTag:attributes:

WAXmlDocument»openTag:	WriteStream»cr
WAXmlDocument»print:	WriteStream»growTo:
WAXmlDocument»urlEncoder	WriteStream»nextPut:
WAXmlDocument»xmlEncoder	WriteStream»nextPutAll:
WAXmlEncoder»nextPutAll:	WriteStream»on:
WeakAnnouncementSubscription class»finalizationList	WriteStream»pastEndPut:
WeakAnnouncementSubscription class»finalizeValue	WriteStream»reset
WeakArray class»finalizationProcess	WriteStream»size
WeakArray class»initialize	WriteStream»space
WeakArray class»restartFinalizationProcess	ZdcAbstractSocketStream class»on:
WeakArray class»startUp:	ZdcAbstractSocketStream»SocketSendData:startingAt:count:
WeakFinalizationList class»hasNewFinalization	ZdcAbstractSocketStream»SocketWaitForData
WeakFinalizationList class»initialize	ZdcAbstractSocketStream»autoFlush:
WeakFinalizationList class»startUp:	ZdcAbstractSocketStream»binary
WeakFinalizationList»swapWithNil	ZdcAbstractSocketStream»bufferSize:
WeakFinalizerItem class»new	ZdcAbstractSocketStream»close
WeakFinalizerItem»add:	ZdcAbstractSocketStream»flush
WeakFinalizerItem»list:object:	ZdcAbstractSocketStream»initializeBuffers
WeakIdentityKeyDictionary»compare:to:	ZdcAbstractSocketStream»initialize
WeakIdentityKeyDictionary»startIndexFor:	ZdcAbstractSocketStream»isBinary
WeakKeyDictionary»associationsDo:	ZdcAbstractSocketStream»nextPut:
WeakKeyDictionary»finalizeValues	ZdcAbstractSocketStream»nextPutAll:
WeakKeyDictionary»fullCheck	ZdcAbstractSocketStream»next
WeakKeyDictionary»grow	ZdcAbstractSocketStream»on:
WeakKeyDictionary»initialize:	ZdcAbstractSocketStream»peek
WeakKeyDictionary»noCheckAdd:	ZdcAbstractSocketStream»shouldSignal:
WeakKeyDictionary»overridingAt:put:	ZdcAbstractSocketStream»socketClose
WeakKeyDictionary»rehash	ZdcAbstractSocketStream»socketIsConnected
WeakKeyDictionary»removeKey:ifAbsent:	ZdcAbstractSocketStream»socketIsDataAvailable
WeakKeyDictionary»scanFor:	ZdcAbstractSocketStream»socketReceiveDataInto:startingAt:count:
WeakKeyDictionary»scanForKeyOrNil:	ZdcAbstractSocketStream»socket
WeakRegistry»add:executor:	ZdcAbstractSocketStream»timeout:
WeakRegistry»add:	ZdcAbstractSocketStream»timeout
WeakRegistry»finalizeValues	ZdcIOBuffer class»on:
WeakRegistry»protected:	ZdcIOBuffer class»onByteArrayOfSize:
WeakRegistry»remove:ifAbsent:	ZdcIOBuffer»advanceWritePointer:
WeakSet»add:	ZdcIOBuffer»availableForReading
WeakSet»do:	ZdcIOBuffer»availableForWriting
WeakSet»growTo:	ZdcIOBuffer»bufferSize
WeakSet»grow	ZdcIOBuffer»buffer
WeakSet»initialize:	ZdcIOBuffer»compact
WeakSet»like:	ZdcIOBuffer»contentsStart
WeakSet»noCheckNoGrowFillFrom:	ZdcIOBuffer»freeSpaceStart
WeakSet»scanFor:	ZdcIOBuffer»isEmpty
WeakSet»scanForEmptySlotFor:	ZdcIOBuffer»isFull
Week class»nameOfDay:	ZdcIOBuffer»next:putAll:startingAt:
WideString»at:	ZdcIOBuffer»nextPut:
WideString»wordAt:	ZdcIOBuffer»next
WriteStream»«	ZdcIOBuffer»on:
WriteStream»contents	ZdcIOBuffer»peek



ZdcIOBuffer»reset	ZnEntityReader»headers:
ZdcSimpleSocketStream»atEnd	ZnEntityReader»headers
ZdcSimpleSocketStream»fillBytes:startingAt:count:	ZnEntityReader»isChunked
ZdcSimpleSocketStream»fillReadBufferNoWait	ZnEntityReader»readEntity
ZdcSimpleSocketStream»fillReadBuffer	ZnEntityReader»stream:
ZdcSimpleSocketStream»flushBytes:startingAt:count:	ZnEntityWriter»headers:
ZdcSimpleSocketStream»flushWriteBuffer	ZnEntityWriter»headers
ZdcSimpleSocketStream»isConnected	ZnEntityWriter»isChunked
ZdcSocketStream»next:putAll:startingAt:	ZnEntityWriter»isGzipped
ZnBivalentWriteStream class»on:	ZnEntityWriter»stream:
ZnBivalentWriteStream»isBinary	ZnEntityWriter»writeEntity:
ZnBivalentWriteStream»nextPut:	ZnHeaders class»defaultResponseHeaders
ZnBivalentWriteStream»nextPutAll:	ZnHeaders class»readFrom:
ZnBivalentWriteStream»on:	ZnHeaders»acceptEntityDescription:
ZnBivalentWriteStream»space	ZnHeaders»at:add:
ZnByteArrayEntity class»bytes:	ZnHeaders»at:ifAbsent:
ZnByteArrayEntity class»designatedMimeType	ZnHeaders»at:put:
ZnByteArrayEntity»bytes:	ZnHeaders»clearContentLength
ZnByteArrayEntity»bytes	ZnHeaders»clearContentType
ZnByteArrayEntity»writeOn:	ZnHeaders»contentLength:
ZnCharacterEncoder class»newForEncoding:	ZnHeaders»contentType:
ZnCharacterEncoder»beLenient	ZnHeaders»headersDo:
ZnCharacterEncoder»encodedByteCountForString:	ZnHeaders»headers
ZnConstants class»defaultHTTPVersion	ZnHeaders»includesKey:
ZnConstants class»defaultMaximumEntitySize	ZnHeaders»isDescribingEntity
ZnConstants class»defaultServerString	ZnHeaders»isEmpty
ZnConstants class»frameworkNameAndVersion	ZnHeaders»normalizeHeaderKey:
ZnConstants class»frameworkName	ZnHeaders»readFrom:
ZnConstants class»frameworkVersion	ZnHeaders»readOneHeaderFrom:
ZnConstants class»httpStatusCodes	ZnHeaders»removeKey:ifAbsent:
ZnConstants class»knownHTTPMethods	ZnHeaders»writeOn:
ZnConstants class»knownHTTPVersions	ZnLineReader class»on:
ZnConstants class»maximumLineLength	ZnLineReader»growBuffer
ZnConstants class»remoteAddressHeader	ZnLineReader»limit:
ZnEntity class»byteArrayEntityClass	ZnLineReader»nextLine
ZnEntity class»bytes:	ZnLineReader»on:
ZnEntity class»new	ZnLineReader»processNext
ZnEntity class»stringEntityClass	ZnLineReader»reset
ZnEntity class»text:	ZnLineReader»store:
ZnEntity class»textCRLF:	ZnLogSupport»debug:
ZnEntity class»type:length:	ZnLogSupport»disable
ZnEntity class»type:	ZnLogSupport»enabled:
ZnEntity»contentLength:	ZnLogSupport»enabled
ZnEntity»contentLength	ZnLogSupport»info:
ZnEntity»contentType:	ZnLogSupport»initialize
ZnEntity»contentType	ZnLogSupport»transaction:
ZnEntityReader»allowsReadingUpToEnd	ZnManagingMultiThreadedServer»closeConnections
ZnEntityReader»binary	ZnManagingMultiThreadedServer»closeSocketStream:
ZnEntityReader»canReadContent	ZnManagingMultiThreadedServer»connections
ZnEntityReader»hasContentLength	ZnManagingMultiThreadedServer»lock

ZnManagingMultiThreadedServer»socketStreamOn:	ZnMultiValueDictionary»at:add:
ZnManagingMultiThreadedServer»stop:	ZnMultiValueDictionary»at:put:
ZnMessage class»readBinaryFrom:UTF8	ZnMultiValueDictionary»checkLimitForKey:
ZnMessage»entity:	ZnMultiValueDictionary»defaultLimit
ZnMessage»entityReaderOn:	ZnMultiValueDictionary»initialize:
ZnMessage»entityWriterOn:	ZnMultiValueDictionary»keysAndValuesDo:
ZnMessage»entity	ZnMultiValueDictionary»limit
ZnMessage»hasEntity	ZnNetworkingUtils class»defaultSocketStreamTimeout
ZnMessage»hasHeaders	ZnNetworkingUtils class»default
ZnMessage»headers:	ZnNetworkingUtils class»ipAddressToString:
ZnMessage»headersDo:	ZnNetworkingUtils class»listenBacklogSize
ZnMessage»headers	ZnNetworkingUtils class»serverSocketOn:
ZnMessage»isConnectionClose	ZnNetworkingUtils class»socketBufferSize
ZnMessage»readBinaryFrom:	ZnNetworkingUtils class»socketStreamOn:
ZnMessage»readHeaderFrom:	ZnNetworkingUtils class»socketStreamTimeout
ZnMessage»setConnectionClose	ZnNetworkingUtils»bufferSize
ZnMessage»wantsConnectionClose	ZnNetworkingUtils»serverSocketOn:
ZnMessage»writeOn:	ZnNetworkingUtils»setServerSocketOptions:
ZnMimeType class»applicationOctetStream	ZnNetworkingUtils»setSocketStreamParameters:
ZnMimeType class»fromString:	ZnNetworkingUtils»socketStreamClass
ZnMimeType class»main:sub:parameters:	ZnNetworkingUtils»socketStreamOn:
ZnMimeType class»main:sub:	ZnNetworkingUtils»timeout
ZnMimeType class»textPlain	ZnNullEncoder class»handlesEncoding:
ZnMimeType»=	ZnPercentEncoder»characterEncoder:
ZnMimeType»asZnMimeType	ZnPercentEncoder»characterEncoder
ZnMimeType»charSet:	ZnPercentEncoder»decode:to:
ZnMimeType»charSet	ZnPercentEncoder»decode:
ZnMimeType»main:	ZnPercentEncoder»encode:to:
ZnMimeType»main	ZnPercentEncoder»encode:
ZnMimeType»parameterAt:ifAbsent:	ZnPercentEncoder»safeSet:
ZnMimeType»parameters:	ZnPercentEncoder»safeSet
ZnMimeType»parameters	ZnRequest»isHttp10
ZnMimeType»printOn:	ZnRequest»method
ZnMimeType»setCharSetUTF8	ZnRequest»readHeaderFrom:
ZnMimeType»sub:	ZnRequest»requestLine:
ZnMimeType»sub	ZnRequest»requestLine
ZnMultiThreadedServer»augmentResponse:forRequest:	ZnRequest»uri
ZnMultiThreadedServer»closeSocketStream:	ZnRequest»wantsConnectionClose
ZnMultiThreadedServer»exceptionSet:	ZnRequestLine class»readFrom:
ZnMultiThreadedServer»executeOneRequestResponse:	ZnRequestLine»isHttp10
ZnMultiThreadedServer»executeRequestResponseLine:	ZnRequestLine»method:
ZnMultiThreadedServer»listenLoop	ZnRequestLine»method
ZnMultiThreadedServer»readRequestBadExceptionSet:	ZnRequestLine»readFrom:
ZnMultiThreadedServer»readRequestSafely:	ZnRequestLine»uri:
ZnMultiThreadedServer»readRequestTerminationExceptionSet:	ZnRequestLine»uri
ZnMultiThreadedServer»serveConnectionsOn:	ZnRequestLine»version:
ZnMultiThreadedServer»workerProcessName	ZnRequestLine»version
ZnMultiThreadedServer»writeResponseBad:on:	ZnResourceMetaUtils class»decodePercent:
ZnMultiThreadedServer»writeResponseSafely:on:	ZnResourceMetaUtils class»encodePercent:safeSet:encoding:
ZnMultiThreadedServer»writeResponseTerminationExceptionSet:	ZnResourceMetaUtils class»parseQueryFrom:

ZnResourceMetaUtils class»queryKeyValueSafeSet	ZnSingleThreadedServer»logResponse:
ZnResourceMetaUtils class»urlPathSafeSet	ZnSingleThreadedServer»log
ZnResourceMetaUtils class»writeQueryFields:on:	ZnSingleThreadedServer»noteAcceptWaitTimedOut
ZnResourceMetaUtils class»writeQueryFields:withText	ZnSingleThreadedServer»periodicTasks
ZnResponse»setConnectionCloseFor:	ZnSingleThreadedServer»process
ZnResponse»setKeepAliveFor:	ZnSingleThreadedServer»readRequest:
ZnResponse»statusLine:	ZnSingleThreadedServer»releaseServerSocket
ZnResponse»statusLine	ZnSingleThreadedServer»serverProcessName
ZnResponse»useConnection:	ZnSingleThreadedServer»serverSocket
ZnResponse»writeOn:	ZnSingleThreadedServer»socketStreamOn:
ZnSeasideServerAdaptorDelegate class»with:	ZnSingleThreadedServer»start
ZnSeasideServerAdaptorDelegate»adaptor:	ZnSingleThreadedServer»stop:
ZnSeasideServerAdaptorDelegate»adaptor	ZnSingleThreadedServer»withMaximumEntitySizeDo:
ZnSeasideServerAdaptorDelegate»handleRequest:	ZnSingleThreadedServer»writeResponse:on:
ZnServer class»defaultServerClass	ZnStatusLine class»badRequest
ZnServer class»initialize	ZnStatusLine class»code:
ZnServer class»managedServers	ZnStatusLine»code:
ZnServer class»on:	ZnStatusLine»code
ZnServer class»shutDown:	ZnStatusLine»reason
ZnServer class»startUp:	ZnStatusLine»version:
ZnServer class»unregister:	ZnStatusLine»version
ZnServer»authenticator	ZnStatusLine»writeOn:
ZnServer»bindingAddress	ZnStringEntity class»designatedMimeType
ZnServer»delegate:	ZnStringEntity class»text:
ZnServer»delegate	ZnStringEntity»computeContentLength
ZnServer»maximumEntitySize	ZnStringEntity»contentLength
ZnServer»optionAt:ifAbsent:	ZnStringEntity»encoder:
ZnServer»optionAt:ifAbsentPut:	ZnStringEntity»encoder
ZnServer»optionAt:put:	ZnStringEntity»hasEncoder
ZnServer»port:	ZnStringEntity»initializeEncoder
ZnServer»port	ZnStringEntity»string:
ZnServer»reader:	ZnStringEntity»string
ZnServer»reader	ZnStringEntity»writeOn:
ZnServer»stop	ZnUTF8Encoder class»handlesEncoding:
ZnServer»unregister	ZnUTF8Encoder class»newForEncoding:
ZnServer»useGzipCompressionAndChunking	ZnUTF8Encoder»decodeBytes:
ZnSignalProgress class»enabled	ZnUTF8Encoder»encodedByteCountFor:
ZnSingleThreadedServer class»default	ZnUTF8Encoder»findFirstNonASCIIIn:startingAt:
ZnSingleThreadedServer class»on:Latin1	ZnUTF8Encoder»next:putAll:startingAt:toStream:
ZnSingleThreadedServer»acceptWaitTimeout	ZnUTF8Encoder»next:putAllASCII:startingAt:toStream:
ZnSingleThreadedServer»augmentResponse:forRequest:	ZnUTF8Encoder»next:putAllByteString:startingAt:toStream:
ZnSingleThreadedServer»authenticateAndDelegateToUse:	ZnUTF8Encoder»nextFromStream:
ZnSingleThreadedServer»authenticateRequest:do:	ZnUTF8Encoder»nextPut:toStream:
ZnSingleThreadedServer»closeDelegate	ZnUnknownHttpMethod class»method:
ZnSingleThreadedServer»handleRequest:	ZnUnknownHttpMethod»method:
ZnSingleThreadedServer»handleRequestProtected:	ZnUrl class»fromString:
ZnSingleThreadedServer»initializeServerSocket	ZnUrl class»schemesNotUsingPath
ZnSingleThreadedServer»isRunning	ZnUrl»addPathSegment:
ZnSingleThreadedServer»logRequest:response:startDate:	ZnUrl»decodePercent:
ZnSingleThreadedServer»logRequest:	ZnUrl»encodePath:on:

ZnUrl»enforceKnownScheme  
ZnUrl»hasFragment  
ZnUrl»hasHost  
ZnUrl»hasPath  
ZnUrl»hasPort  
ZnUrl»hasQuery  
ZnUrl»hasScheme  
ZnUrl»hasUsername  
ZnUrl»isSchemeUsingPath  
ZnUrl»parseFrom:defaultScheme:  
ZnUrl»parseFrom:  
ZnUrl»parsePath:  
ZnUrl»printAuthorityOn:  
ZnUrl»printOn:  
ZnUrl»printPathOn:  
ZnUrl»printPathQueryFragmentOn:  
ZnUrl»printQueryOn:  
ZnUrl»query:  
ZnUrl»query  
ZnUrl»scheme  
ZnUtils class»httpDate:  
ZnUtils class»httpDate  
ZnUtils class»nextPutAll:on:  
ZnUtils class»signalProgress:total:  
ZnUtils class»streamingBufferSize  
ZnZincServerAdaptor»basicStart  
ZnZincServerAdaptor»configureDelegate  
ZnZincServerAdaptor»configureServerForBinaryReading  
ZnZincServerAdaptor»defaultCodec  
ZnZincServerAdaptor»defaultDelegate  
ZnZincServerAdaptor»defaultZnServer  
ZnZincServerAdaptor»isRunning  
ZnZincServerAdaptor»isStopped  
ZnZincServerAdaptor»requestAddressFor:  
ZnZincServerAdaptor»requestBodyFor:  
ZnZincServerAdaptor»requestCookiesFor:  
ZnZincServerAdaptor»requestFieldsFor:  
ZnZincServerAdaptor»requestHeadersFor:  
ZnZincServerAdaptor»requestMethodFor:  
ZnZincServerAdaptor»requestUrlFor:  
ZnZincServerAdaptor»requestVersionFor:  
ZnZincServerAdaptor»responseFrom:  
ZnZincServerAdaptor»server  
ZnZincServerAdaptor»shutDown  
ZnZincServerAdaptor»startUp