Extended results of 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. Extended results of Tornado: A Run-Fail-Grow approach for Dynamic Application Tayloring. [Research Report] Inria. 2014. hal-00996908v3

HAL Id: hal-00996908
https://hal.archives-ouvertes.fr/hal-00996908v3
Submitted on 15 Jul 2014

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.
Extended results of Tornado

A Run-Fail-Grow approach

for

Dynamic Application Tayloring

Author: Guillermo POLITO
Version: 1.1

Supervisors: Stéphane DUCASSE
Noury BOURAQADI
Luc FABRESE

July 15, 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.

In this report, we present the results we obtained from using 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

1  Used Methodology 3
2  Hello World Application 3
3  Seaside Web Application 3

A  Appendix: Method List of a Nurtured Hello World Application 6
B  Appendix: Entry Points to Tailor the Seaside Web Application 8
C  Appendix: Method List of Seaside Counter Application with Full Pharo Seed 9
D  Appendix: Method List of Seaside Counter Application with Empty Seed 15
1 Used 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 option¹. The results prove the soundness of our solution.

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 1 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 1: Entry point of the Hello World application with an empty seed.

<table>
<thead>
<tr>
<th>Size(KB)</th>
<th>Occupied(%)</th>
<th>Saved(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>12872</td>
<td>100%</td>
</tr>
<tr>
<td>Tailored</td>
<td>131</td>
<td>1%</td>
</tr>
</tbody>
</table>

Table 1: Results of the tailored Hello World application.

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.

¹Pharo allows to prepare a snapshot for production. This option cleans some caches and removes some well known objects from the system, thus, freeing space.
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>
<thead>
<tr>
<th></th>
<th>Size (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref. Pharo Base Libraries (P)</td>
<td>12872</td>
</tr>
<tr>
<td>Ref. Seaside Framework (S)</td>
<td>4326</td>
</tr>
<tr>
<td>Ref. Counter Application (R)</td>
<td>52</td>
</tr>
<tr>
<td><strong>Total Ref. Application (P+S+C)</strong></td>
<td><strong>17250</strong></td>
</tr>
</tbody>
</table>

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.

Table 2 shows the results obtained when tailoring this application with each of these two seeds. Figure 2 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”.

Figure 2: Tailoring Map. Tailoring map describing the Seaside application generated with the empty seed (left) and the full Pharo seed (right).

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
References

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Class/Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array class-&gt;new:</td>
<td>FileStream=openOnHandle:name:forWrite:Standard</td>
</tr>
<tr>
<td>ArrayedCollection=size</td>
<td>FileStream=primWrite:from:starting:forWrite:Standard</td>
</tr>
<tr>
<td>Association class=key:value:</td>
<td>GreekEnvironment class=supportedLanguages</td>
</tr>
<tr>
<td>Association=value</td>
<td>HashTableSizes class=atLeast</td>
</tr>
<tr>
<td>Association=value</td>
<td>HashTableSizes class=sizes</td>
</tr>
<tr>
<td>BlockClosure=on:do:</td>
<td>HashedCollection class=newProto</td>
</tr>
<tr>
<td>BlockClosure=repeat</td>
<td>HashedCollection=atNewIndex:put:</td>
</tr>
<tr>
<td>BlockClosure=valueNoContextSwitch</td>
<td>HashedCollection=findElementOrNil:</td>
</tr>
<tr>
<td>ByteString class-&gt;compare:with:collated:</td>
<td>HashedCollection=fullCheck</td>
</tr>
<tr>
<td>ByteString class-&gt;stringFirstInString:inSet:startingAt:</td>
<td>HashedCollection=grow</td>
</tr>
<tr>
<td>ByteString-&gt;at:put:</td>
<td>Integer=asCharacter</td>
</tr>
<tr>
<td>ByteString-&gt;at:</td>
<td>JapaneseEnvironment class=supportedLanguages</td>
</tr>
<tr>
<td>ByteString-&gt;isByteString</td>
<td>KoreanEnvironment class=supportedLanguages</td>
</tr>
<tr>
<td>ByteString-&gt;replaceFrom:to:with:startingAt:</td>
<td>LanguageEnvironment class=currentPlatform</td>
</tr>
<tr>
<td>ByteString class-&gt;unicodeToByteTable</td>
<td>LanguageEnvironment class=defaultSystemConverter</td>
</tr>
<tr>
<td>ByteString class-&gt;nextPut:to:Stream:</td>
<td>LanguageEnvironment class=initKnownEnvironments</td>
</tr>
<tr>
<td>ByteString class-&gt;unicodeToByte:</td>
<td>LanguageEnvironment class=knownEnvironments</td>
</tr>
<tr>
<td>Character class=\cr</td>
<td>LanguageEnvironment class/localeID:</td>
</tr>
<tr>
<td>Character class=\lf</td>
<td>LanguageEnvironment/localeID:</td>
</tr>
<tr>
<td>Character class=value:</td>
<td>Latin1Environment class=supportedLanguages</td>
</tr>
<tr>
<td>Character=\n</td>
<td>Latin2Environment class=supportedLanguages</td>
</tr>
<tr>
<td>Character=asInteger</td>
<td>Latin9Environment class=supportedLanguages</td>
</tr>
<tr>
<td>Character=asciiValue</td>
<td>Latin9Environment class=systemConverterClass</td>
</tr>
<tr>
<td>Character=charCode</td>
<td>Locale class=currentPlatform</td>
</tr>
<tr>
<td>Collection-&gt;detect:ifNone:</td>
<td>Locale class=determineCurrentLocale</td>
</tr>
<tr>
<td>Dictionary-&gt;at:ifAbsent:</td>
<td>Locale=determineLocaleID</td>
</tr>
<tr>
<td>Dictionary-&gt;at:ifPresent:</td>
<td>Locale=determineLocale</td>
</tr>
<tr>
<td>Dictionary-&gt;at:put:</td>
<td>Locale=fetchISO2Language</td>
</tr>
<tr>
<td>Dictionary-&gt;noCheckAdd:</td>
<td>Locale=languageEnvironment</td>
</tr>
<tr>
<td>Dictionary-&gt;scanFor:</td>
<td>Locale/localeID:</td>
</tr>
<tr>
<td>FileStream class-&gt;newForStdio</td>
<td>Locale/localeID:</td>
</tr>
<tr>
<td>FileStream class-&gt;new</td>
<td>Locale=primCountry</td>
</tr>
<tr>
<td>FileStream class-&gt;standardIOStreamNamed:forWriteLocale=primLanguage</td>
<td>Locale=isoLanguage:isoCountry:</td>
</tr>
<tr>
<td>FileStream class=startUp:</td>
<td>LocaleID class=isoLanguage:isoCountry</td>
</tr>
<tr>
<td>FileStream class=stdioHandles</td>
<td>LocaleID class=isoLanguage:</td>
</tr>
<tr>
<td>FileStream class=stdoutCharacter=isCharacter</td>
<td>LocaleID class=isoString:</td>
</tr>
<tr>
<td>FileStream class=voidStdioFiles</td>
<td>LocaleID=hash</td>
</tr>
<tr>
<td>FileStream=collectionSpeciesStandard</td>
<td>LocaleID=hash</td>
</tr>
<tr>
<td>FileStream=enableReadBufferingSmalltalkImage=vmLocaleID=isoCountry</td>
<td>LocaleID=isoLanguage:isoCountry:</td>
</tr>
<tr>
<td>FileStream=isBinaryStandard</td>
<td>LocaleID=isoLanguage:isoCountry:</td>
</tr>
<tr>
<td>FileStream=next:putAll:startingAt:Standard</td>
<td>LocaleID=isoLanguage:</td>
</tr>
<tr>
<td>FileStream=nextPut:Standard</td>
<td>LookupKey class=key:</td>
</tr>
</tbody>
</table>


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 3) only consists in starting the web server as the base libraries are initialized and available in the seed. The latter one (Figure 4) includes the initialization of the minimal runtime needed to do networking.

```
ZnZincServerAdaptor startOn: 8888.
```

Figure 3: Entry point of the Seaside application with a full Pharo seed.

```
"We initialize some classes of the system"
SmalltalkImage initializeForTornado.
Symbol initializeForTornado.
Object initialize.
ExternalSemaphoreTable initialize.
Socket initialize.
Delay initialize.
Delay startUp: true.
Delay shutDown: true.
OSPlatform initialize.
DiskStore initialize.
FileStream initialize.
NetNameResolver initialize.
DateAndTime initialize.
ProcessorScheduler initialize.
WeakFinalizationList initialize.
UUIDGenerator initialize.
WeakArray initialize.
GRPharoRandomProvider initialize.
WASlime initialize.
UIManager basicDefault: DummyUIManager new.
ZnServer initialize.
WAServerManager initialize.
Smalltalk instVarNamed: 'session' put: Smalltalk newSessionObject.
Smalltalk startupImage: true snapshotWorked: true.
"Finally we start the web server"
ZnZincServerAdaptor startOn: 8888.
```

Figure 4: 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
WAAttributeSearchContext initializeWithKey:
WAAccessIntervalReapingStrategy initialize
WAAttributeSearchContext isAttributeInheritedOn:
WAAccessIntervalReapingStrategy interval
WAAttributeSearchContext isAttributeLocalOn:
WAAccessIntervalReapingStrategy reaping
WAAttributeSearchContext key
WAAccessIntervalReapingStrategy stored:key:
WABrush initialize
WAActionCallback block:
WABrush parent
WAActionCallback evaluateWithArgument:
WABrush with:
WAActionCallback isEnabledFor:
WABrجرedirect:
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
WAAdmin class=defaultServerManager
WACache initializeCollections
WAAdmin class=serverAdaptors
WACache initializeMutex
WAAnchorTag callback:
WACache initialize
WAAnchorTag tag
WACache keyAtValue:ifAbsent:
WAAnchorTag url
WACache keyAtValue:
WAAnchorTag with:
WACache keySize
WAApplication contentType
WACache missStrategy
WAApplication doesHandlerSupportCookies:
WACache notifyRemoved:key:
WAApplication handleDefault:
WACache notifyRetrieved:key:
WAApplication handleFiltered:
WACache notifyStored:key:
WAApplication isApplication
WACache pluginsDo:
WAApplication isImplemented:
WACache reapingStrategy
WAApplication keyField
WACache reap
WAApplication libraries
WACache removalAction
WAApplication mainClass
WACache setExpiryPolicy:
WAApplication mimeType
WACache setMissStrategy:
WAApplication newSession
WACache setReapingStrategy:
WAApplication resourceBaseUrl
WACache setRemovalAction:
WAApplication sessionClass
WACache store:
WAApplicationConfiguration parents
WACacheCapacityConfiguration describeOn:
WAAttributeSearchContext class:key:target:
WACacheMissStrategy missed:
WAAttributeSearchContext at:ifPresent:
WACachePlugin configuration
WAAttributeSearchContext at:put:
WACachePlugin defaultConfiguration
WAAttributeSearchContext attribute
WACachePlugin initialize
WAAttributeSearchContext cachedValues
WACachePlugin removed:key:
WAAttributeSearchContext findAttributeAndSelectAncestorsOf:
WAHtmlRoot>initialize
WAHtmlRoot>meta
WAHtmlRoot>openOn:
WAHtmlRoot>title:
WAHtmlRoot>writeElementsOn:
WAHtmlRoot>writeFootOn:
WAHtmlRoot>writeHeadOn:
WAHtmlRoot>writeScriptsOn:
WAHtmlRoot>writeStylesOn:
WAHttpVersion class>fromString:
WAHttpVersion class>major:minor:
WAHttpVersion class>readFrom:
WAHttpVersion class>initializeWithMajor:minor:
WAInitialRequestVisitor class>request:
WAInitialRequestVisitor>initializeWithRequest:
WAInitialRequestVisitor>request:
WAInitialRequestVisitor>visitPresenter:
WAKeyGenerator class>current:
WAKeyGenerator>keyOfLength:
WALastAccessExpiryPolicy>defaultConfiguration:
WALastAccessExpiryPolicy>initialize:
WALastAccessExpiryPolicy>isExpired:key:
WALastAccessExpiryPolicy>retrieved:key:
WALastAccessExpiryPolicy>timeout:
WALeastRecentlyUsedExpiryPolicy>defaultConfiguration:
WALeastRecentlyUsedExpiryPolicy>initialize:
WALeastRecentlyUsedExpiryPolicy>isExpired:key:
WALeastRecentlyUsedExpiryPolicy>maximumAge:
WALeastRecentlyUsedExpiryPolicy>removed:key:
WALeastRecentlyUsedExpiryPolicy>retrieved:key:
WALeastRecentlyUsedExpiryPolicy>stored:key:
WAMergedRequestFields class>on:
WAMergedRequestFields>allAt:
WAMergedRequestFields>at:ifAbsent:
WAMergedRequestFields>includesKey:
WAMergedRequestFields>initializeOn:
WAMergedRequestFields>keysAndValuesDo:
WAMergedRequestFields>keysDo:
WAMetaElement>content:
WAMetaElement>contentScriptType:
WAMetaElement>contentType:
WAMetaElement>encodeBeforeOn:
WAMetaElement>responseHeaderName:
WAMetaElement>tag:
WAMimeType>fromString:
WAMimeType>main:sub:
WAMimeType>textJavascript:
WAMimeType>textPlain:
WAMimeType>charset:
WARenderContext.visitor: WARenderRequest.setHeaders:
WARenderContext.visitor: WARenderRequest.setPostFields:
WARenderLoopConfiguration.parent: WARenderRequest.setRemoteAddress:
WARenderLoopContinuation.createActionContinuation: WARenderRequest.url:
WARenderLoopContinuation.createRenderContinuation: WARenderRequest.url:
WARenderLoopContinuation.presenter: WARenderRequestContext.class.request.response.codec:
WARenderLoopContinuation.toPresenterSendRoot: WARenderRequestContext.application
WARenderLoopContinuation.updateRoot: WARenderRequestContext.charSet
WARenderLoopContinuation.updateStates: WARenderRequestContext.codec
WARenderLoopContinuation.updateUrl: WARenderRequestContext.consumer
WARenderLoopContinuation.withNotificationHandler: WARenderRequestContext.destroy
WARenderLoopMain.createRoot: WARenderRequestContext.handlers
WARenderLoopMain.prepareRoot: WARenderRequestContext.handler
WARenderLoopMain.rootClass: WARenderRequestContext.handler
WARenderLoopMain.rootDecorationClasses: WARenderRequestContext.newDocument
WARenderLoopMain.start: WARenderRequestContext.push.during:
WARenderPhaseContinuation.createHtmlRootWithWARenderRequestContext.registry
WARenderPhaseContinuation.createRenderContext: WARenderRequestContext.request
WARenderPhaseContinuation.handleRequest: WARenderRequestContext.respond:
WARenderPhaseContinuation.processRendering: WARenderRequestContext.respondGenerator
WARenderVisitor.class.context: WARenderRequestContext.response
WARenderVisitor.initializeWithContext: WARenderRequestContext.responseGenerator
WARenderVisitor.renderContext: WARenderRequestContext.session
WARenderVisitor.visitPainter: WARenderFilter.handleFiltered:
WARenderer.class.context: WARenderFilter.initialize
WARenderer.actionUrl: WARenderFilter.next
WARenderer.callbacks: WARenderFilter.setNext:
WARenderer.context: WARenderFilter.updateStates:
WARenderer.document: WARenderHandler.addFilter:
WARenderer.flush: WARenderHandler.addFilterLast:
WARenderer.initializeWithContext: WARenderHandler.basicUrl
WARenderer.render: WARenderHandler.configuration:
WARenderer.text: WARenderHandler.configuration
WARequest.class.method.uri.version: WARenderHandler.defaultConfiguration
WARequest.at: ifAbsent: WARenderHandler.documentClass
WARequest.cookiesAt: WARenderHandler.filters
WARequest.cookies: WARenderHandler.filter
WARequest.destroy: WARenderHandler.handle:
WARequest.fields: WARenderHandler.initialize
WARequest.headerAt: ifAbsent: WARenderHandler.isApplication
WARequest.headerAt: WARenderHandler.isRoot
WARequest.initializeWithMethod.uri.version: WARenderHandler.parent
WARequest.isGet: WARenderHandler.preferenceAt:
WARequest.isPrefetch: WARenderHandler.responseGenerator
WARequest.isXmlHttpRequest: WARenderHandler.serverHostname
WARequest.method: WARenderHandler.serverPath
WARequest.postFields: WARenderHandler.serverPort
WARequest.queryFields: WARenderHandler.serverProtocol
WARequest.setBody: WARenderHandler.setFilter:
WARequest.setCookies: WARenderHandler.setParent:
WA Tag Brush » open Tag
WA Tag Brush » store Callback:
WA Tag Brush » with:
WA Tag Canvas » space
WA Unescaped Document » initialize With Stream: codec:
WA Update Root Visitor class » root:
WA Update Root Visitor » initialize With Root:
WA Update Root Visitor » root:
WA Update Root Visitor » visit Painter:
WA Update States Visitor class » snapshot:
WA Update States Visitor » initialize With Snapshot:
WA Update States Visitor » snapshot:
WA Update States Visitor » visit Presenter:
WA Update Url Visitor class » url:
WA Update Url Visitor » initialize With Url:
WA Update Url Visitor » url:
WA Update Url Visitor » visit Painter:
WA Update Url Visitor » visit:
WA Update Url Visitor » url:
WA Xml Document » close Tag:
WA Xml Document » destroy
WA Url class » absolute:
WA Url class » decode Percent:
WA Url add All To Path:
WA Url add Field » value:
WA Url add Field:
WA Url add To Path:
WA Url decode:
WA Url decoded With:
WA Url encode On:
Zn Seaside Server Adaptor Delegate class » with:
WA Url encode Path On:
Zn Seaside Server Adaptor Delegate » adaptor:
WA Url encode Query On:
Zn Seaside Server Adaptor Delegate » adaptor
WA Url encode Scheme And Authority On:
Zn Seaside Server Adaptor Delegate » handle Request:
WA Url fragment
Zn Zinc Server Adaptor » basic Start
WA Url initialize From String:
Zn Zinc Server Adaptor » configure Delegate
WA Url initialize
Zn Zinc Server Adaptor » configure Server For Binary Reading
WA Url is Seaside Field:
Zn Zinc Server Adaptor » default Codec
WA Url parse Path:
Zn Zinc Server Adaptor » default Delegate
WA Url parse Query:
Zn Zinc Server Adaptor » default Zinc Server
WA Url password
Zn Zinc Server Adaptor » is Stopped
WA Url path:
Zn Zinc Server Adaptor » request Address For:
WA Url path Elements In: do:
Zn Zinc Server Adaptor » request Body For:
WA Url path:
Zn Zinc Server Adaptor » request Cookies For:
WA Url post Copy
Zn Zinc Server Adaptor » request Fields For:
WA Url print On:
Zn Zinc Server Adaptor » request Headers For:
WA Url query Fields:
Zn Zinc Server Adaptor » request Method For:
WA Url query Fields
Zn Zinc Server Adaptor » request Url For:
WA Url seaside Url
Zn Zinc Server Adaptor » request Version For:
WA Url slash:
Zn Zinc Server Adaptor » response From:
WA Url sub Strings In: split By: do:
Zn Zinc Server Adaptor » server
WA Url user
WA Url Encoder class » on: codec:
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:
Array»isSelfEvaluating
Array»printOn:
Array»replaceFrom:to:with:startingAt:
ArrayedCollection class»new:withAll:
ArrayedCollection class»new:
ArrayedCollection class»with:with:with:
ArrayedCollection class»with:with:
ArrayedCollection class»with:
ArrayedCollection»mergeSortFrom:to:by:
ArrayedCollection»size
Association class»key:value:
Association class»key:value:
Association class»key:value:WeakKey
Association class»expiredWeakKey
Association class»key:WeakKey
Association class»key:value:WeakKey
Association class»value:WeakKey
Association class»key:WeakKey
Association class»value:
Association class»valueWeakKey
Association class»value:
BlockClosure»argumentCount
BlockClosure»asContextWithSender:
BlockClosure»asContext
BlockClosure»assert
BlockClosure»cull:
BlockClosure»ensure:
BlockClosure»fixCallbackTemps
BlockClosure»forkAt:named:
BlockClosure»forkAt:
BlockClosure»ifCurtailed:
BlockClosure»ifError:
BlockClosure»isClosure
BlockClosure»newProcess
BlockClosure»numArgs
BlockClosure»numCopiedValues
BlockClosure»outerContext

ByteArray»asByteArray
ByteArray»replaceFrom:to:with:startingAt:
ByteString class»compare:with:collated:
ByteString class»findFirstInString:inSet:startingAt:
ByteString class»indexOfAscii:inString:startingAt:
ByteString class»stringHash:initialHash:
ByteString class»translate:from:to:table:
ByteSymbol class»stringHash:initialHash:
ByteSymbol»at:
ByteSymbol»findSubstring:in:startingAt:matchTable:
ByteSymbol»isByteString
ByteSymbol»isOctetString
ByteSymbol»privateAt:put:
ByteSymbol»species
ChangesLog class»default
ChangesLog»recordStartupStamp
Character class»codePoint:
Character class»cr
Character class»lf
Character class » space
Character class » value:
Character » =
Character » asCharacter
Character » asInteger
Character » asSymbol
Character » asUppercase
Character » asciiValue
Character » charCode
Character » characterSet
Character » codePoint
Character » digitValue
Character » greaselnteger
Character » isAlphaNumeric
Character » isCharacter
Character » isDigit
Character » isLetter
Character » isOctetCharacter
Character » isSeparator
Character » isVowel
Character » leadingChar
Character » to:
Collection class » withAll:
Collection » addAll:
Collection » allSatisfy:
Collection » anySatisfy:
Collection » asArray
Collection » detect: ifNone:
Collection » emptyCheck
Collection » inject: into:
Collection » isCollection
Collection » isEmptyOrNil
Collection » notEmpty
Collection » printElementsOn:
Collection » printNameOn:
Collection » printOn:
Collection » removeAll:
Collection » removeAllFoundIn:
Collection » sorted:
Collection » sorted
CommandLineUIManager class » replacing:
CommandLineUIManager » initialize
CommandLineUIManager » replacing:
CompiledMethod » frameSize
CompiledMethod » header
CompiledMethod » initialPC
CompiledMethod » isPrimitive
CompiledMethod » numLiterals
Date class » fromSeconds:
Date class » fromString:
Date class » readFrom:
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»isActiveOf:
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»suspendedList
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»relinquishProcessorForMicroseconds:
SequenceableCollection»do:
ProcessorScheduler class»startUp
SequenceableCollection»doWithIndex:
ProcessorScheduler»activePriority
SequenceableCollection»first
ProcessorScheduler»activeProcess
SequenceableCollection»first
ProcessorScheduler»highPIOpriority
SequenceableCollection»from:to:put:
ProcessorScheduler»highestPriority
SequenceableCollection»grownBy:
ProcessorScheduler»lowPIOpriority
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
Time»nanoSecond
Time»print24:showSeconds:on:
Time»printOn:
Time»seconds
Time»tick:
TimeZone»offset
Timespan class»starting:duration:
Timespan»<
Timespan»dayOfMonth
Timespan»duration:
Timespan»month
Timespan»start:
Timespan»start
Timespan»year
True»ifFalse:
True»not
True»|
UILevel class»basicDefault:
UILevel class»default:WAUnesappedDocument»initialize:
UILevel class»default
UILevel»activate
UILevel»beDefault
UILevel»boot:during:
UILevel»deactivate
UILevel»onSnapshot:
UTF16TextConverter class»encodingNames
UTF8DecomposedTextConverter class»encodingNames
UUIDGenerator class»initialize
UUIDGenerator class»startUp
UndefinedObject»encodeOn:
UndefinedObject»isNil
UndefinedObject»notNil
UndefinedObject»seasideUrl
UndefinedObject»shallowCopy
Unicode class»isDigit:
Unicode class»isLetter:
Unicode class»toUppercase:
VirtualMachine class»allocationsBetweenGC:
VirtualMachine class»getSystemAttribute:
VirtualMachine class»initialize:
VirtualMachine class»interpreterClass
VirtualMachine class»interpreterSourceDate
VirtualMachine class»isPharoVM
VirtualMachine class»isRunningCogit
VirtualMachine class»maxExternalSemaphores
VirtualMachine class»parameterAt:put:
VirtualMachine class»parameterAt:
VirtualMachine class»setGCPARAMS:
VirtualMachine class»tenuringThreshold:
ZnManagingMultiThreadedServer>socketStreamOn:
ZnManagingMultiThreadedServer>stop:
ZnMessage class>readBinaryFrom:UTF8
ZnMessage>entity:
ZnMessage>entityReaderOn:
ZnMessage>entityWriterOn:
ZnMessage>entity
ZnMessage>hasEntity
ZnMessage>hasHeaders
ZnMessage>headers:
ZnMessage>headersDo:
ZnMessage>headers
ZnMessage>isConnectionClose
ZnMessage>readBinaryFrom:
ZnMessage>readHeaderFrom:
ZnMessage>setConnectionClose
ZnMessage>wantsConnectionClose
ZnMessage>writeOn:
ZnMimeType class>applicationOctetStream
ZnMimeType>asZnMimeType
ZnMimeType= ZnMimeType>charSet:
ZnMimeType>charSet
ZnMimeType>main:
ZnMimeType>parameterAt:ifAbsent:
ZnMimeType>parameters:
ZnMimeType>printOn:
ZnMimeType>setCharSetUTF8
ZnMimeType>sub:
ZnMimeType>sub:
ZnMultiThreadedServer>augmentResponse:forRequest:
ZnMultiThreadedServer>closeSocketStream:
ZnMultiThreadedServer>exceptionSet:
ZnMultiThreadedServer>executeOneRequestResponseOn:
ZnMultiThreadedServer>executeRequestResponseLoopOn:
ZnMultiThreadedServer>listenLoop
ZnMultiThreadedServer>readRequestBadExceptionSet
ZnMultiThreadedServer>readRequestSafely:
ZnMultiThreadedServer>serveConnectionsOn:
ZnMultiThreadedServer>workerProcessName
ZnMultiThreadedServer>writeResponseBad:on:
ZnMultiThreadedServer>writeResponseSafely:on:
ZnMultiThreadedServer>writeResponseTerminationExceptionSet
ZnNetworkingUtils class>defaultSocketStreamTimeout
ZnNetworkingUtils class>ipAddressToString:
ZnNetworkingUtils class>listenBacklogSize:
ZnNetworkingUtils class>serverSocketOn:
ZnNetworkingUtils class>socketBufferSize:
ZnNetworkingUtils class>socketStreamOn:
ZnNetworkingUtils class>socketStreamTimeout:
ZnNetworkingUtils>bufferSize
ZnNetworkingUtils>setServerSocketOptions:
ZnNetworkingUtils>setSocketStreamParameters:
ZnPercentEncoder>characterEncoder:
ZnPercentEncoder<charSet: ZnPercentEncoder>decode:
ZnPercentEncoder>encode:
ZnPercentEncoder>encode:
ZnResourceMetaUtils class>decodePercent:
ZnResourceMetaUtils class>encodePercent:safeSet:encoding:
ZnRequest>isHttp10
ZnRequest>method
ZnRequest>version
ZnRequest>uri
ZnRequestLine>isHttp10
ZnRequestLine>method
ZnRequestLine>uri
ZnRequestLine>version
ZnResourceMetaUtils class>parseQueryFrom: