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 Tailoring

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>Size (KB)</th>
<th></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 (C)</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
(CPER) 2007-2013’, the Cutter ANR project, ANR-10-BLAN-0219.

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.

- Array class->new:
- ArrayedCollection->size
- Association class->key:value:
- Association->value:
- BlockClosure->on:do:
- BlockClosure->repeat
- BlockClosure->valueNoContextSwitch
- ByteString class->compare:with:collated:
- ByteString class->findFirstInString:inSet:startingAt:
- ByteString class->stringHash:initialHash:
- ByteString->at:put:
- ByteString->at:
- ByteString->isByteString
- ByteString->replaceFrom:to:with:startingAt:
- ByteString->at:
- ByteString->isByteString
- ByteString->replaceFrom:to:with:startingAt:
- ByteString class->unicodeToByteTable
- ByteString->nextPut:toStream:
- ByteString->unicodeToByteTable:
- Character class->or
- Character class->asInteger
- Character->asciiValue
- Character->charCode
- Collection->detect:ifNone:
- Dictionary->at:ifAbsent:
- Dictionary->at:ifPresent:
- Dictionary->at:put:
- Dictionary->noCheckAdd:
- Dictionary->scanFor:
- FileStream class->newForStdio
- FileStream class->new
- FileStream class->standardIOStreamNamed:forWriteStandard
- FileStream class->startUp:
- FileStream class->stdioHandles
- FileStream class->stdoutCharacter
- FileStream class->voidStdioFiles
- FileStream class->collectionSpeciesStandard
- FileStream class->enableReadBufferingSmalltalkImage
- FileStream->openOnHandle:name:forWrite:Standard
- FileStream->primWrite:from:startingAt:count:Standard
- GreekEnvironment class->supportedLanguages
- HashedCollection class->newProto
- HashedCollection->atNewIndex:put:
- HashedCollection->findElementOrNil:
- HashedCollection->fullCheck
- HashedCollection->grow
- HashedCollection->initialize:
- Integer->asCharacter
- JapaneseEnvironment class->supportedLanguages
- KoreanEnvironment class->supportedLanguages
- LanguageEnvironment class->currentPlatform
- LanguageEnvironment class->defaultSystemConverter
- LanguageEnvironment class->initKnownEnvironments
- LanguageEnvironment class->knownEnvironments
- LanguageEnvironment class->localeID:
- LanguageEnvironment class->localeID
- Locale class->currentPlatform
- Locale class->determineCurrentLocale
- Locale class->determineLocaleID
- Locale class->determineLocale
- Locale class->fetchISO2Language
- Locale class->languageEnvironment
- Locale class->localeID:
- Locale class->localeID
- LocaleID class->isCountry:isCountry:
- LocaleID class->isCountry:
- LocaleID class->isoString:
- LocaleID class->hash
- LocaleID->hash
- LocaleID->isoCountry:
- LocaleID->isoCountry:
- LocaleID->isoLanguage:
- LocaleID->isoLanguage
- LookupKey class->key:
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.
## 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.

- `WAActionCallback` block:
  - `evaluateWithArgument:`
  - `isEnabledFor:`
  - `signalRenderNotification`
- `WAActionPhaseContinuation` continue:
  - `handleRequest`
  - `renderContext:`
  - `runCallbacks`
  - `shouldRedirect`
- `WAApplication` contentType:
  - `doesHandlerSupportCookies:`
  - `handleDefault:`
  - `handleFiltered:`
  - `isApplication`
  - `isImplemented:`
  - `keyField`
  - `libraries`
  - `mainClass`
  - `mimeType`
  - `newSession`
  - `resourceBaseUrl`
  - `sessionClass`
- `WAApplicationConfiguration` parents:
  - `configuration`
  - `removeAction`
  - `setExpirationPolicy:`
  - `setMissStrategy:`
  - `setReapingStrategy:`
  - `setRemovalAction:`
  - `store`
- `WAAttributeSearchContext` key:
  - `initializeWithKey:`
  - `isAttributeInheritedOn:`
  - `isAttributeLocalOn:`
- `WABrush` initialize:
  - `parent`
  - `setParent:canvas:`
- `WABufferedResponse` contents:
  - `destroy`
  - `initializeOn:`
  - `stream`
- `WACache` at:
  - `expiryPolicy`
  - `initialize`
  - `initializeCollections`
  - `initializeMutex`
  - `initializeReapingStrategy`
  - `reap`
  - `reapStrategy`
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:
WACoponent->accept:
WACoponent->acceptDecorated:
WACoponent->decoration
WACoponent->initialize
WACoponent->updateStates:
WAConfigurationDescription->add:to:
WAConfigurationDescription->addAttribute:
WAConfigurationDescription->attributes
WAConfigurationDescription->expressions
WAConfigurationDescription->initialize
WAConfigurationDescription->integer:
WAConfirmedRequestFilter->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->uriFor:
WADocument class->on:codec:
WADocument->close
WADocument->destroy
WADocument->initializeWithStream: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
Array class
Array
Array
ArrayedCollection class
ArrayedCollection class
ArrayedCollection class
ArrayedCollection class
ArrayedCollection class
ArrayedCollection class
ArrayedCollection class
ArrayedCollection class
ArrayedCollection class
Association class
Association class
Association
Association
Association
Association
Association
Association
Association
Association
Association
Association
Association
Association
Association
Association
Association
Association
Association
Association
Association
Association
Association
Association
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
BlockClosure
ByteString class
ByteString class
ByteString class
ByteString class
ByteString class
ByteSymbol class
ByteSymbol class
ChangesLog class
ChangesLog class
Character class
Character class
Character class
Character class
Character class
Character class
Character class
Character class
Character class
Character class
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 greaseliInteger
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 isNotEmpty
Collection noneSatisfy:
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
CompiledMethod numTemps
CompiledMethod objectAt:
CompiledMethod primitive
CompiledMethod replacing:
CompiledMethod=initialize
CompiledMethod=initialize
CompiledMethod=initialize
CompiledMethod=initialize
CompiledMethod=initialize
ContextPart contextEnsure:
ContextPart contextOn:do:
ContextPart newForMethod:
ContextPart theReturnMethod
ContextPart activateMethod:withArgs.receiver.class:
ContextPart activateReturn:value:
ContextPart at:put:
ContextPart at:
ContextPart bottomContext
ContextPart cut:
ContextPart doPop
ContextPart exceptionClass
ContextPart exceptionHandlerBlock
ContextPart exceptionHandlerIsActive:
ContextPart exceptionHandlerIsActive
ContextPart findContextSuchThat:
ContextPart findNextHandlerContextStarting
ContextPart findNextUnwindContextUpTo:
ContextPart handleSignal:
ContextPart insertSender:
ContextPart isDead
ContextPart jump
ContextPart methodReturnTop
ContextPart nextHandlerContext
ContextPart pop
ContextPart privSender:
ContextPart push:
ContextPart pushTemporaryVariable:
ContextPart releaseTo:
ContextPart resume:through:
ContextPart resume:
ContextPart return:from:
ContextPart return:through:
ContextPart return:
ContextPart runUntilErrorOrReturnFrom:
ContextPart runUntilSignal:
ContextPart singleRelease
ContextPart stackp:
ContextPart stepToCallee
ContextPart step
ContextPart terminate:
ContextPart terminateTo:
Date class=fromSeconds:
Date class=fromString:
Date class=next
Date class=starting:
Date class=year.month.day:
Dictionary+includesKey: Exception class+«
Dictionary+initialize:GRSmall Exception class+handles:
Dictionary+initializeSmall Exception class+signal:
Dictionary+isEmpty:GRSmall Exception class+signal:
Dictionary+keyAtValue:ifAbsent: Exception+description
Dictionary+keysAndValuesDo:SmallDictionary Exception+isResumable
Dictionary+isEmpty GRSmall class+new
Dictionary+keysAndValuesDo: Exception+messageText
Dictionary+keysAndValuesDo: Exception+printOn:
Dictionary+keysDo:GRSmall Exception+privHandlerContext:
Dictionary+noCheckAdd: Exception+receiver
Dictionary+postCopy:GRSmall 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+seasideRequestFields:GRSmall ExceptionSet+add:
Dictionary+sizeGRSmall ExceptionSet+handles:
Dictionary+valuesDo: ExceptionSet+initialize
DiskStore+checkVMVersion ExtendedNumberParser+allowPlusSign
DiskStore+initialize ExtendedNumberParser+nextNumber
DiskStore+reset ExternalSemaphoreTable+clearExternalObjects
DiskStore+shutdown: ExternalSemaphoreTable+collectionBasedOn:withRoomFor:
DiskStore+startup: ExternalSemaphoreTable+ FreedSlotsIn:ratherThanIncreaseSizeTo:
DiskStore+useFilePlugin ExternalSemaphoreTable+initialize
Duration class=days:hours:minutes:seconds:nanoSeconds ExternalSemaphoreTable+registerExternalObject:
Duration class=days:hours:minutes:seconds: ExternalSemaphoreTable+ safelyRegisterExternalObject:
Duration class=days: ExternalSemaphoreTable+slotFor:
Duration class=seconds:nanoSeconds: ExternalSemaphoreTable+unprotectedExternalObjects:
Duration class=seconds: ExternalSemaphoreTable+unprotectedExternalObjects
Duration++ False+not
Duration+=Duration 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+initialize
Duration+=seconds:nanoSeconds: FileStream+shutDown:
Duration+=ticks FileStream+startUp:
DynamicVariable+class=value:duration: FileStream+stdioHandles
DynamicVariable+value:duration: FileStream+voidStdioFiles
EUCJPTextConverter+class=encodingNames Float+precision
EUCKRTTextConverter+class=encodingNames Float+adaptToInteger:andSend:
EncodedCharSet+class=charsetAt: Float+asFloat
EventManager+class=actionMaps Float+isInfinite
EventManager+class=flushEvents Float+timesTwoPower:
Socket>sendSomeData:startIndex:count:for:
Socket>sendSomeData:startIndex:count:
Socket>setOption:value:
Socket>socketHandle
Socket>unregister
Socket>waitForAcceptFor:
Socket>waitForConnectionFor:ifTimedOut:
Socket>waitForDataFor:
Socket>waitForDataFor:ifClosed:ifTimedOut:
Socket>waitForDisconnectionFor:
Socket>waitForSendDoneFor:
SparseLargeTable>at:
SparseLargeTable>noCheckAt:
SparseLargeTable>pvtCheckIndex:
SparseLargeTable>size
SqNumberParser>allowPlusSign
SqNumberParser>makeIntegerOrScaledInteger
SqNumberParser>readScale
Stream>basicNext
Stream>nextPutAll:
Stream>print:
String class>crlf
String class>empty
String class>=
String class>asString
String class>asZnMimeType
String class>asZnUrl
String class>forEncoding:
String class>initialize
TextConverter class>allEncodingNames
TextConverter class>encodingNames
TextConverter class>forEncoding:
TextConverter class>initialize
TextConverter class>nextFromStream:UTF8
Time class>dateAndTimeFromSeconds:
Time class>dateAndTimeNow
Time class>fromSeconds:
Time class>hour:minute:second:nanoSecond:
Time class>hour24
Time class>hour
Time class>minute
Time class>millisecondClockValue
Time class>milliseconds:since:
Time class>milliseconds:since:
Time class>milliseconds:since:
Time class>milliseconds:since:
Time class>millisecondsWhenClockTicks
Time class>primSecondsClock
Time class>readFrom:
Time class>seconds:nanoSeconds:
Time class>seconds:whenClockTicks
Time class>totalSeconds
Time>hour24
Time>hour
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+|WAActionPhaseContinuation/shouldRedirect |
| True=|WADefaultServerManager |
| UIManager class/basicDefault: | WADefaultServerAdapters |
| UIManager class/default:WAUnescapedDocument | WAActionWithSignalBlock: |
| UIManager class/default | WAnchorTag/tag |
| UIManager=activate | WAnchorTag/url |
| UIManager=beDefault | WAnchorTag/with: |
| UIManager=boot:during: | WAApplication/contentType |
| UIManager=deactivate | WAApplication/doesHandleSupportCookies: |
| 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: | WAApplication/Configuration/parents |
| Unicode class/toUppercase: | WAttributeSearchContext/class/key/target: |
| VirtualMachine class/allocationBetweenGC: | WAttributeSearchContext/class/at:ifPresent: |
| VirtualMachine class/getSystemAttribute: | WAttributeSearchContext/class/at:put: |
| VirtualMachine class/interpreterClass | WAttributeSearchContext/class/attribute |
| VirtualMachine class/interpreterSourceDate | WAttributeSearchContext/class/cachedValues |
| VirtualMachine class/interpreterSourceVersion | WAttributeSearchContext/class/findAttributeAndSelectAncestorsOf: |
| VirtualMachine class=isPharoVM | WAttributeSearchContext/class/initializeWithKey: |
| VirtualMachine class=isRunningCogit | WAttributeSearchContext/class/isAttributeInheritedOn: |
| VirtualMachine class=maxExternalSemaphores | WAttributeSearchContext/class/isAttributeLocalOn: |
| VirtualMachine class/parameterAt:put: | WAttributeSearchContext/class/key |
| VirtualMachine class/parameterAt: | WAttributeSearchContext/class/parent |
| VirtualMachine class/setGCPARAMeters | WAttributeSearchContext/class/setParent:canvas: |