Extended results of Tornado: A Run-Fail-Grow approach for Dynamic Application Tailoring
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 Tailoring. [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.

![Figure 1: Entry point of the Hello World application with an empty seed.](image)

<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 (R)</td>
<td>52</td>
</tr>
<tr>
<td>Total Ref. Application (P+S+C)</td>
<td>17250</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Size (KB)</th>
<th>Occupied (%)</th>
<th>Saved (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P'+S'+C' / P+S+C</td>
<td>573</td>
<td>3%</td>
</tr>
<tr>
<td>P''+S''+C'' / P+S+C</td>
<td>13090</td>
<td>76%</td>
</tr>
<tr>
<td>S''+C'' / S+C</td>
<td>218</td>
<td>5%</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.

Array class»new:
ArrayedCollection»size
Association class»key:value:
Association»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»putAll:
ByteString»isCharacter
JapaneseEnvironment class»supportedLanguages
KoreanEnvironment class»supportedLanguages
LanguageEnvironment class»currentPlatform
Latin1Environment class»supportedLanguages
Latin2Environment class»supportedLanguages
Latin9Environment class»systemConverterClass
Locale»localeID:
LocaleID class»isoString:
LocaleID»hash
LocaleID»isoLanguage:isoCountry:
LocaleID»isoCountry:
LocaleID»hash
LocaleID»isLanguage:isoCountry:
LocaleID»isCountry:localeID:
LocaleID»localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
localeID:
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.

```smalltalk
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
WAAccessIntervalReapingStrategy»initialize
WAAccessIntervalReapingStrategy»interval
WAAccessIntervalReapingStrategy»reap
WAAccessIntervalReapingStrategy»stored:key:
WAActionCallback»block:
WAActionCallback»evaluateWithArgument:
WAActionCallback»isEnabledFor:
WAActionCallback»signalRenderNotification
WAActionPhaseContinuation»continue
WAActionPhaseContinuation»handleRequest
WAActionPhaseContinuation»renderContext:
WAActionPhaseContinuation»runCallbacks
WAActionPhaseContinuation»shouldRedirect
WAAdmin class»defaultServerManager
WAAdmin class»serverAdaptors
WAAdmin class»defaultServerManager
WAAnchorTag»callback:
WAAnchorTag»tag
WAAnchorTag»url
WAAnchorTag»with:
WAApplication»contentType
WAApplication»doesHandlerSupportCookies:
WAApplication»handleDefault:
WAApplication»handleFiltered:
WAApplication»isApplication
WAApplication»isImplemented:
WAApplication»keyField
WAApplication»libraries
WAApplication»mainClass
WAApplication»mimeType
WAApplication»newSession
WAApplication»resourceBaseUrl
WAApplication»sessionClass
WAApplicationConfiguration»parents
WAAttributeSearchContext class»key:target:
WAAttributeSearchContext»isAttributeInheritedOn:
WAAttributeSearchContext»key:
WAAttributeSearchContext»isAttributeLocalOn:
WAAttributeSearchContext»initialize
WAAttributeSearchContext»initializeWithKey:
WAAttributeSearchContext»isAttributeInheritedOn:
WAAttributeSearchContext»initializeWithKey:
WAAttributeSearchContext»isAttributeInheritedOn:
WAAttributeSearchContext»initializeWithKey:
WARequestHandler
WAResponse class
WAResponse
WAResponse cookies
WAResponse destroy
WAResponse found
WAResponse headerAt:ifAbsent:
WAResponse headers
WAResponse initializeOn:
WAResponse initialize:
WAResponse location:
WAResponse redirectTo:
WAResponse status:message:
WAResponse status:
WAResponseGenerator class
WAResponseGenerator expiredRegistryKey
WAResponseGenerator initializeOn:
WAResponseGenerator requestContext
WAResponseGenerator request
WAResponseGenerator respond
WAResponseGenerator response
WARoot class
WARoot setContext:
WAScriptGenerator initialize
WAScriptGenerator loadScripts
WAScriptGenerator writeLoadScriptsOn:
WAScriptGenerator writeScriptTag:on:
WAServerAdaptor class
WAServerAdaptor defaultPort
WAServerAdaptor defaultRequestHandler
WAServerAdaptor handle:
WAServerAdaptor handlePadding:
WAServerAdaptor handleRequest:
WAServerAdaptor initializeWithManager:
WAServerAdaptor initialize:
WAServerAdaptor manager
WAServerAdaptor port:
WAServerAdaptor process:
WAServerAdaptor requestFor:
WAServerAdaptor responseFor:
WAServerAdaptor start
WAServerManager class
WAServerManager adaptor:
WAServerManager canStart:
WAServerManager register:
WAServerManager start:
WASession actionField
WASession actionUrlForContinuation:
WASession actionUrlForKey:
WASession application
WASession clearJumpTo
WASession createCache
WASession handleFiltered:
WASession initializeFilters
WASession initialize:
WASession isSession
WASession presenter
WASession start:
WASession unknownRequest
WASession updateRoot:
WASession updateStates:
WASession updateUrl:
WASessionContinuation basicValue
WASessionContinuation captureAndInvoke
WASessionContinuation captureState
WASessionContinuation redirectToContinuation:
WASessionContinuation registerForUrl:
WASessionContinuation registereForUrl:
WASessionContinuation request
WASessionContinuation respond:
WASessionContinuation setStates:
WASessionContinuation states
WASessionContinuation updateStates:
WASessionContinuation updateUrl:
WASessionContinuation value
WASessionContinuation withUnregisteredHandlerDo:
WASnapshot initialize
WASnapshot register:
WASnapshot reset
WASnapshot restore
WATagBrush after
WATagBrush attributes
WATagBrush before
WATagBrush closeTag
WATagBrush isClosed
WATagBrush isClosed
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»repeatWithGClIf:
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:with:  BlockClosure»valueWithPossibleArguments:
ArrayedCollection class»with:  ByteArray»asByteArray
ArrayedCollection»mergeSortFrom:to:by:  ByteString»compare:with:collated:
ArrayedCollection»sort:  ByteString»findFirstInString:inSet:startingAt:
Association class»key:value:  ByteString»indexOfAscii:inString:startingAt:
Association class»key:value:  ByteString»stringHash:initialHash:
Association»expireWeakKey  ByteString»translate:from:to:table:
Association»expiredWeakKey  ByteString»asByteArray
Association»key:WeakKey  ByteString»at:
Association»key: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»isHexString
BlockClosure»asContextWithSender:  ByteString»isOctetString
BlockClosure»asContext  ByteString»replaceFrom:to:startingAt:
BlockClosure»assert  ByteString»stringHash:initialHash:
BlockClosure»cull:  ByteString»at:
BlockClosure»ensure:  ByteString»findSubstring:in:startingAt:matchTable:
BlockClosure»fixCallbackTemps  ByteString»isHexString
BlockClosure»forkAt:named:  ByteString»privateAt:put:
BlockClosure»forkAt:  ByteString»species
BlockClosure»ifCurtailed:  ByteString»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»fl
Float
Fraction numerator:denominator:
Fraction >=
Fraction reduced
Fraction setNumerator:denominator:
Fraction truncated
GRCodecStream on:
GRCodecStream atEnd
GRCodecStream initializeOn:
GRCodecStream initializeOn:
GRCodecStream initializeOn:
GRCodecStream supportsEncoding:
GRCodecStream encoderFor:
GRCodecStream url
GRNullCodecStream nextPutAll:
GRObject new
GRObject initialize
GROrderedMultiMap allAt:
GROrderedMultiMap at:add:
GRPharoConverterCodecStream on:converter
GRPharoConverterCodecStream contents
GRPharoConverterCodecStream initializeOn:converter
GRPharoConverterCodecStream size
GRPharoGenericCodec supportedEncodingNames
GRPharoGenericCodec supportsEncoding:
GRPharoLatin1Codec supportedEncodingNames
GRPharoLatin1Codec supportsEncoding:
GRPharoPlatform addToShutDownList:
GRPharoPlatform addToStartupList:
GRPharoPlatform includesUnsafeUrlCharacter:
GRPharoPlatform includesUnsafeXmlCharacter:
GRPharoSemaphoreClass
GRPharoRandomProvider initialize
GRPharoRandomProvider nextInt:
GRPharoRandomProvider randomClass
GRPharoUtf8Codec basicForEncoding:
GRPharoUtf8Codec supportsEncoding:
GRPharoUtf8Codec decode:
GRPharoUtf8Codec decoderFor:
GRPharoUtf8Codec encoderFor:
GRPharoUtf8Codec name
GRPharoUtf8Codec url
GRPharoUtf8Codec encodeFast:
GRPharoUtf8Codec next:
GRPharoUtf8Codec nextPut:
GRPharoUtf8Codec nextPutAll:
GRPlatform current
GRPlatform reducedConflictDictionary
GRSmallDictionary new
GRSmallDictionary new
PositionableStream•isEmpty
PositionableStream•on:
PositionableStream•originalContents
PositionableStream•peekFor:
PositionableStream•peek
PositionableStream•position:
PositionableStream•position
PositionableStream•reset
PositionableStream•skip:
PositionableStream•skipSeparators
PositionableStream•skipTo:
Process class•forContext:priority:
Process•activateReturn:value:
Process•calleeOf:
Process•complete:
Process•isActiveOf:
Process•name:
Process•popTo:
Process•primitiveResume
Process•priority:
Process•priority
Process•psValueAt:put:
Process•psValueAt:
Process•resume
Process•return:value:
Process•suspendedContext:
Process•suspendingList
Process•suspend
Process•terminate
ProcessLocalVariable•class•value:
ProcessLocalVariable•value:
ProcessSpecificVariable•class•soleInstance
ProcessSpecificVariable•class•value
ProcessSpecificVariable•default
ProcessSpecificVariable•value
ProcessorScheduler class•idleProcess
ProcessorScheduler class•initialize
ProcessorScheduler class•relinquishProcessorForMicroseconds:
ProcessorScheduler class•startUp
ProcessorScheduler•activePriority
ProcessorScheduler•activeProcess
ProcessorScheduler•highOPriority
ProcessorScheduler•highestPriority
ProcessorScheduler•lowOPriority
ProcessorScheduler•lowestPriority
ProcessorScheduler•terminateActive
ProcessorScheduler•timingPriority
ProcessorScheduler•userInterruptPriority
ProtoObject•basicIdentityHash
ProtoObject•flag:
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:
WACachePlugin»store:
WACacheRegistry»advanceKey
WACacheRegistry»handle:
WACacheRegistry»initialize
WACacheRegistry»nextKey
WACacheRegistry»store:
WACanvas»brush:
WACanvas+flush
WACanvas»nest:
WACanvas»render:
WACanvas»text:
WACanvas»accept:
WACanvas»acceptDecorated:
WACanvas»decoration
WACanvas»initialize
WACanvas»updateStates:
WAConfigurationDescription»add:to:
WAConfigurationDescription»addAttribute:
WAConfigurationDescription»attributes
WAConfigurationDescription»expressions
WAConfigurationDescription»initialize
WAConfigurationDescription»integer:
WAConfiguredRequestFilter»configuration
WACounter»count:
WACounter»increase
WACounter»decrease
WACounter»initialize
WACounter»states
WADefaultScriptGenerator»close:on:
WADefaultScriptGenerator»open:on:
WADispatcher»nameOfHandler:
WADispatcher»urlFor:
WADispatcher class»default
WADispatcher class»parents
WADispatcher»handlerAt:ifAbsent:
WADispatcher»handlerAt:with:
WADispatcher»handlers
WADispatcher class»on:codec:
WADispatcher class»on:
WADocument class»on:codec:
WADocument class»on:
WADocument»close:
WADocument»destroy
WADocument»initializeWithStream:codec:
WADocument»nextPut:
WADocument»nextPutAll:
WADocument»open:
WADynamicVariable class»use:to:with:
WADynamicVariable class»value
WAEncoder class»on:table:
WAEncoder class»on:
WAEncoder class»on:
WAEncoder class»initializeOn:table:
WAEncoder»nextPut:
WAErrorLog class»exceptionSelector
WAExampleComponent»rendererClass
WAExceptionHandler class»context:
WAExceptionFilter»exceptionHandler
WAExceptionFilter»exceptionHandler
WAExceptionFilter»handleFiltered:
WAExceptionFilter»handleFiltered:
WAExceptionFilter»handleFiltered: