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:

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 option\(^1\). 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.

\[^1\] 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.

![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.
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>Component</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>
<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
(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:
Association » value
BlockClosure » on: do:
BlockClosure » repeat
BlockClosure » value NoContextSwitch
ByteString class » compare: with: collated:
ByteString class » find first in String: in Set: starting at:
ByteString » at: put:
ByteString » at:
ByteString » is ByteString
ByteString » replace from: to: with: starting at:
ByteString » unicode to Byte Table
ByteTextConverter class » unicode to Byte Table
ByteTextConverter » next put: to Stream:
ByteTextConverter » unicode to Byte:
Character class » cr
Character class » equals
Character class » as Integer
Character class » ascii Value
Character class » char Code
Collection » detect: if None:
Dictionary » at: if Absent:
Dictionary » at: if Present:
Dictionary » put:
Dictionary » no Check Add:
Dictionary » scan For:
FileStream class » new for Stdio
FileStream class » new
FileStream class » standard IO Stream Named: for Write
FileStream class » start Up:
FileStream class » std io Handles
FileStream class » std out Character » is Character
FileStream class » void Std io Files
FileStream » collection Species Standard
FileStream » enable Read Buffering Small talk Image: » String
FileStream » is Binary Standard
FileStream » next put All: starting at:
FileStream » next Put: Standard
locale ID » hash
locale ID » locale ID:
locale ID » iso Language: iso Country:
locale ID » iso Language: iso Country:
locale ID » iso Language: iso Country:
locale ID » iso Country:
locale ID » iso Language: iso Country:
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.
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
WAApplication::configuration::parents
WAAttributeSearchContext class::key:target:
WAAttributeSearchContext::at:ifPresent:
WAAttributeSearchContext::at:put:
WAAttributeSearchContext::attribute
WAAttributeSearchContext::cachedValues
WAAttributeSearchContext::findAttributeAndSelectAncestorsOf:
WAAttributeSearchContext::initializeWithKey:
<table>
<thead>
<tr>
<th>Class/Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAHtmlRoot::initialize</td>
<td>Initialize the HTML root object.</td>
</tr>
<tr>
<td>WAHtmlRoot::meta</td>
<td>Define the HTML meta information.</td>
</tr>
<tr>
<td>WAHtmlRoot::openOn</td>
<td>Open the HTML document.</td>
</tr>
<tr>
<td>WAHtmlRoot::title</td>
<td>Set the HTML document title.</td>
</tr>
<tr>
<td>WAHtmlRoot::writeElementsOn</td>
<td>Write the HTML elements.</td>
</tr>
<tr>
<td>WAHtmlRoot::writeFootOn</td>
<td>Write the HTML footer.</td>
</tr>
<tr>
<td>WAHtmlRoot::writeHeadOn</td>
<td>Write the HTML head.</td>
</tr>
<tr>
<td>WAHtmlRoot::writeScriptsOn</td>
<td>Write the HTML scripts.</td>
</tr>
<tr>
<td>WAHtmlRoot::writeStylesOn</td>
<td>Write the HTML styles.</td>
</tr>
<tr>
<td>WAMimeType::greaseString</td>
<td>Define the MIME type for grease string.</td>
</tr>
<tr>
<td>WAMimeType::main</td>
<td>Define the MIME type for main.</td>
</tr>
<tr>
<td>WAMimeType::sub</td>
<td>Define the MIME type for sub.</td>
</tr>
<tr>
<td>WAMimeType::parameters</td>
<td>Define the MIME type for parameters.</td>
</tr>
<tr>
<td>WAHttpVersion::fromString</td>
<td>Convert a string to an HTTP version.</td>
</tr>
<tr>
<td>WAHttpVersion::major:minor</td>
<td>Define the major and minor version numbers.</td>
</tr>
<tr>
<td>WAHttpVersion::readFrom</td>
<td>Read an HTTP version.</td>
</tr>
<tr>
<td>WAHttpVersion::initialize</td>
<td>Initialize the HTTP version.</td>
</tr>
<tr>
<td>WAHttpVersion::initializeWithMajor:minor</td>
<td>Initialize the HTTP version with major and minor numbers.</td>
</tr>
<tr>
<td>WAInitialRequestVisitor::request</td>
<td>Define the initial request visitor.</td>
</tr>
<tr>
<td>WAInitialRequestVisitor::initializeWithRequest</td>
<td>Initialize the initial request visitor with request.</td>
</tr>
<tr>
<td>WAInitialRequestVisitor::request</td>
<td>Define the request visitor.</td>
</tr>
<tr>
<td>WAInitialRequestVisitor::requestVisitor</td>
<td>Define the request visitor visitor.</td>
</tr>
<tr>
<td>WAInitialRequestVisitor::visitPresenter</td>
<td>Define the visit presenter.</td>
</tr>
<tr>
<td>WAKeyGenerator::keyOfLength</td>
<td>Define the key generator.</td>
</tr>
<tr>
<td>WAKeyGenerator::keyOfLength:proxy</td>
<td>Define the key generator with proxy.</td>
</tr>
<tr>
<td>WALastAccessExpiryPolicy::defaultConfiguration</td>
<td>Define the default configuration for last access expiry policy.</td>
</tr>
<tr>
<td>WALastAccessExpiryPolicy::initialize</td>
<td>Initialize the last access expiry policy.</td>
</tr>
<tr>
<td>WALastAccessExpiryPolicy::isExpired:key</td>
<td>Check if the key is expired.</td>
</tr>
<tr>
<td>WALastAccessExpiryPolicy::maximumAge</td>
<td>Define the maximum age for the last access expiry policy.</td>
</tr>
<tr>
<td>WALastAccessExpiryPolicy::removed:key</td>
<td>Remove the key from the last access expiry policy.</td>
</tr>
<tr>
<td>WALastAccessExpiryPolicy::retrieved:key</td>
<td>Retrieve the key from the last access expiry policy.</td>
</tr>
<tr>
<td>WALastAccessExpiryPolicy::timeout</td>
<td>Define the timeout for the last access expiry policy.</td>
</tr>
<tr>
<td>WALeastRecentlyUsedExpiryPolicy::defaultConfiguration</td>
<td>Define the default configuration for least recently used expiry policy.</td>
</tr>
<tr>
<td>WALeastRecentlyUsedExpiryPolicy::initialize</td>
<td>Initialize the least recently used expiry policy.</td>
</tr>
<tr>
<td>WALeastRecentlyUsedExpiryPolicy::isExpired:key</td>
<td>Check if the key is expired.</td>
</tr>
<tr>
<td>WALeastRecentlyUsedExpiryPolicy::maximumAge</td>
<td>Define the maximum age for the least recently used expiry policy.</td>
</tr>
<tr>
<td>WALeastRecentlyUsedExpiryPolicy::removed:key</td>
<td>Remove the key from the least recently used expiry policy.</td>
</tr>
<tr>
<td>WALeastRecentlyUsedExpiryPolicy::retrieved:key</td>
<td>Retrieve the key from the least recently used expiry policy.</td>
</tr>
<tr>
<td>WALeastRecentlyUsedExpiryPolicy::timeout</td>
<td>Define the timeout for the least recently used expiry policy.</td>
</tr>
<tr>
<td>WAMergedRequestFields::on</td>
<td>Define the merged request fields object.</td>
</tr>
<tr>
<td>WAMergedRequestFields::allAt:</td>
<td>Define the merged request fields with allAt.</td>
</tr>
<tr>
<td>WAMergedRequestFields::atIfAbsent:</td>
<td>Define the merged request fields with atIfAbsent.</td>
</tr>
<tr>
<td>WAMergedRequestFields::includesKey:</td>
<td>Define the merged request fields with includesKey.</td>
</tr>
<tr>
<td>WAMergedRequestFields::initializeOn:</td>
<td>Initialize the merged request fields object.</td>
</tr>
<tr>
<td>WAMergedRequestFields::keysAndValuesDo:</td>
<td>Define the merged request fields with keysAndValuesDo.</td>
</tr>
<tr>
<td>WAMergedRequestFields::keysDo:</td>
<td>Define the merged request fields with keysDo.</td>
</tr>
<tr>
<td>WAMimeType::charset</td>
<td>Define the MIME type for charset.</td>
</tr>
<tr>
<td>WAMimeType::textJavascript</td>
<td>Define the MIME type for text Javascript.</td>
</tr>
<tr>
<td>WAMimeType::textPlain</td>
<td>Define the MIME type for text Plain.</td>
</tr>
<tr>
<td>WAPainter::renderWithContext</td>
<td>Render the painter with context.</td>
</tr>
<tr>
<td>WAPainter::updateRoot</td>
<td>Update the painter.</td>
</tr>
<tr>
<td>WAPainter::updateUrl</td>
<td>Update the painter URL.</td>
</tr>
<tr>
<td>WAPainterVisitor::visitComponent</td>
<td>Visit the painter component.</td>
</tr>
<tr>
<td>WAPainterVisitor::visitDecorationsOfComponent</td>
<td>Visit the painter decorations of component.</td>
</tr>
<tr>
<td>WAPainterVisitor::visitPainter</td>
<td>Visit the painter.</td>
</tr>
<tr>
<td>WAPathConsumer::path</td>
<td>Define the path consumer.</td>
</tr>
<tr>
<td>WAPathConsumer::atEnd</td>
<td>Define the path consumer at end.</td>
</tr>
<tr>
<td>WAPathConsumer::upToEnd</td>
<td>Define the path consumer up to end.</td>
</tr>
<tr>
<td>WAPathConsumer::visit</td>
<td>Define the path consumer visit.</td>
</tr>
<tr>
<td>WAPathConsumer::childrenDo</td>
<td>Define the path consumer childrenDo.</td>
</tr>
<tr>
<td>WAPathConsumer::children</td>
<td>Define the path consumer children.</td>
</tr>
<tr>
<td>WAPathConsumer::initialRequest:</td>
<td>Define the path consumer initial request.</td>
</tr>
<tr>
<td>WAPathConsumer::script</td>
<td>Define the path consumer script.</td>
</tr>
<tr>
<td>WAPathConsumer::style</td>
<td>Define the path consumer style.</td>
</tr>
<tr>
<td>WARegistryConfiguration::parents</td>
<td>Define the registry configuration parents.</td>
</tr>
<tr>
<td>WAPresenter::updateRoot</td>
<td>Update the presenter.</td>
</tr>
<tr>
<td>WAPresenter::updateStates</td>
<td>Update the presenter states.</td>
</tr>
<tr>
<td>WAPresenter::updateUrl</td>
<td>Update the presenter URL.</td>
</tr>
<tr>
<td>WAPresenter::childrenDo</td>
<td>Define the presenter childrenDo.</td>
</tr>
<tr>
<td>WAPresenter::children</td>
<td>Define the presenter children.</td>
</tr>
<tr>
<td>WAPresenter::initialRequest:</td>
<td>Define the presenter initial request.</td>
</tr>
<tr>
<td>WAPresenter::script</td>
<td>Define the presenter script.</td>
</tr>
<tr>
<td>WAPresenter::style</td>
<td>Define the presenter style.</td>
</tr>
<tr>
<td>WAPresenterGuide::client</td>
<td>Define the presenter guide client.</td>
</tr>
<tr>
<td>WAPresenterGuide::initializeWithClient</td>
<td>Initialize the presenter guide with client.</td>
</tr>
<tr>
<td>WAPresenterGuide::visit</td>
<td>Define the presenter guide visit.</td>
</tr>
<tr>
<td>WAPresenterGuide::visitPainter</td>
<td>Define the presenter guide visit painter.</td>
</tr>
<tr>
<td>WARenderContext::actionBaseUrl:</td>
<td>Define the render context action base URL.</td>
</tr>
<tr>
<td>WARenderContext::actionUrl:</td>
<td>Define the render context action URL.</td>
</tr>
<tr>
<td>WARenderContext::callbacks</td>
<td>Define the render context callbacks.</td>
</tr>
<tr>
<td>WARenderContext::defaultVisitor</td>
<td>Define the render context default visitor.</td>
</tr>
<tr>
<td>WARenderContext::destroy</td>
<td>Define the render context destroy.</td>
</tr>
<tr>
<td>WARenderContext::initialize</td>
<td>Initialize the render context.</td>
</tr>
<tr>
<td>WARenderContext::resourceUrl:</td>
<td>Define the render context resource URL.</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
</tr>
</tbody>
</table>
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: 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 class»with:with:with: ArrayedCollection class»with:with:with: ArrayedCollection class»mergeSortFrom:to:by: ArrayedCollection»size ArrayedCollection»sort: Association class»key:value: Association class»key:value: Association class»key:value: WeakKey Association»expireWeakKey Association»expiredWeakKey Association»key:WeakKey Association»key:value:WeakKey Association»keyWeakKey Association»value:WeakKey Association»value: Association»value:WeakKey Association»value: Association»value:WeakKey BlockClosure»argumentCount BlockClosure»asContextWithSender: BlockClosure»asContext BlockClosure»assert BlockClosure»cull: BlockClosure»ensure: BlockClosure»fixCallbackTemps BlockClosure»forkAt:named: BlockClosure»forkAt: BlockClosure»ifCurtailed: BlockClosure»ifError: BlockClosure»asClosure BlockClosure»newProcess BlockClosure»on:do: BlockClosure»on:fork: BlockClosure»outerContext ByteArray»asByteArray ByteString»at:put: ByteString»at: ByteString»beginsWith: ByteString»byteAt:put: ByteString»byteSize ByteString»findSubstring:in:startingAt:matchTable: ByteString»findSubstringViaPrimitive:in:startingAt:matchTable: ByteString»isByteString ByteString»isOctetString ByteString»replaceFrom:to:with:startingAt: ByteString»stringHash:initialHash: ByteString»translate:from:to:table: ByteSymbol class»findSubstring:in:startingAt:matchTable: ByteSymbol»at: ByteSymbol»privateAt:put: ByteSymbol»species ByteSymbol»string: ByteSymbol»findSubstring:in:startingAt:matchTable: ByteSymbol»isByteString ByteSymbol»isHexString: ByteString»byteAt:put: CNGB TextConverter class»encodingNames CP1250 TextConverter class»encodingNames CP1253 TextConverter class»encodingNames ChangesLog class»default ChangesLog class»recordStartupStamp Character class»codePoint: Character class»cr Character class»lf
Dictionary includesKey:
Dictionary initializeGRSmall
Dictionary initializeSmall
Dictionary isEmptyGRSmall
Dictionary keyAtValue:ifAbsent:
Dictionary keysAndValuesDo: SmallDictionary
Dictionary new
Dictionary keysAndValuesDo:
Dictionary keysAndValuesDo:
Dictionary keysDo: GRSmall
Dictionary noCheckAdd:
Dictionary postCopyGRSmall
Dictionary privateAt: put: GRSmall
Dictionary privateAt: put: Small
Dictionary rehash
Dictionary removeKey: ifAbsent:
Dictionary scanFor:
Dictionary seasideRequestFieldsGRSmall
Dictionary sizeGRSmall
Dictionary valuesDo:
DiskStore class checkVMVersion
DiskStore class initialize
DiskStore class reset
DiskStore class shutDown:
DiskStore class setUp:
DiskStore class useFilePlugin
Duration class days: hours: minutes: seconds: nanoSeconds:
Duration class days: hours: minutes: seconds:.nanoSeconds:
Duration class days:
Duration class seconds: nanoSeconds:
Duration class seconds: nanoSeconds:
Duration class seconds:
Duration++
Duration--
Duration asDuration
Duration asMilliSeconds
Duration asNanoSeconds
Duration asSeconds
Duration days
Duration isZero
Duration negated
Duration seconds: nanoSeconds:
Duration ticks
DynamicVariable class + value: during:
DynamicVariable value: during:
EUCJPTextConverter class + encodingNames
EUCKRTextConverter class + encodingNames
EncodedCharSet class charsetAt:
EventManager class + actionMaps
EventManager class + flushEvents
Exception class:
Exception class + handles:
Exception class + signal:
Exception class + signal:
Exception + description
Exception + isResumable
Exception + messageText
Exception + printOn:
Exception + privHandlerContext:
Exception + resume:
Exception + resumeUnchecked:
Exception + signal:
Exception + signalerContext
ExceptionSet add:
ExceptionSet handles:
ExceptionSet initialize
ExtendedNumberParser allowPlusSign
ExtendedNumberParser nextNumber
ExternalSemaphoreTable class clearExternalObjects
ExternalSemaphoreTable class collectionBasedOn: withRoomFor:
ExternalSemaphoreTable class freedSlotsIn: ratherThanIncreaseSizeTo:
ExternalSemaphoreTable class + initialize
ExternalSemaphoreTable class + registerExternalObject:
FileStream class flushAndVoidStdioFiles
FileStream class initialize
FileStream class shutDown:
FileStream class setUp:
FileStream class stdioHandles
FileStream class voidStdioFiles
Float class + precision
Float adaptToInteger: andSend:
Float asFloat
Float isInfinite
Float timesTwoPower:
Float truncated
Fraction numerator: denominator:
Fraction: >=
Fraction: reduced
Fraction: setNumerator: denominator:
Fraction: truncated
GRCodecStream on:
GRCodecStream: atEnd
GRCodecStream: initializeOn:
GRCodecStream: initializeOn: converter:
GRCodecStream: size
GRCodecStream: nextPutAll:
GRCodecStream: next:
GRCodecStream: nextPut:
GRCodecStream: nextPutAll:
GRNullCodec codecName
GRNullCodec supportsEncoding:
GRNullCodec encoderFor:
GRNullCodec url
GRNullCodecStream nextPutAll:
GROrderedMultiMap allAt:
GROrderedMultiMap at:add:
GRPharoConverterCodecStream on: converter
GRPharoConverterCodecStream contents
GRPharoConverterCodecStream initializeOn: converter
GRPharoConverterCodecStream size
GRPharoConverterCodecStream initializeOn:
GRPharoConverterCodecStream size
GRPharoGenericCodec supportedEncodingNames
GRPharoGenericCodec supportsEncoding:
GRPharoLatin1Codec supportedEncodingNames
GRPharoLatin1Codec supportsEncoding:
GRPharoPlatform addToShutDownList:
GRPharoPlatform addToList:
GRPharoPlatform includesUnsafeUrlCharacter:
GRPharoPlatform includesUnsafeXmlCharacter:
GRPharoPlatform semaphoreClass
GRPharoRandomProvider initialize
GRPharoRandomProvider nextInt:
GRPharoRandomProvider randomClass
GRPharoUtf8Codec basicForEncoding:
GRPharoUtf8Codec supportsEncoding:
GRPharoUtf8Codec decode:
GRPharoUtf8Codec decoderFor:
GRPharoUtf8Codec encoderFor:
GRPharoUtf8Codec name
GRPharoUtf8Codec url
GRPharoUtf8CodecStream encodeFast:
GRPharoUtf8CodecStream next:
GRPharoUtf8CodecStream nextPut:
GRPharoUtf8CodecStream nextPutAll:
GRPlatform current
GRPlatform reducedConflictDictionary
GRSmallDictionary new:
GRSmallDictionary new
HashTableSizes atLeast:
HashTableSizes sizes
HashedCollection new:
HashedCollection new:
HashedCollection sizeFor:
HashedCollection array
HashedCollection atNewIndex: put:
HashedCollection findElementOrNil:
HashedCollection fullCheck
HashedCollection grow
HashedCollection size
Heap withAll: sortBlock:
Heap do:
Heap growHeap reSort
Heap growTo:
Heap isEmpty:
Heap privateRemoveAt:
Heap remove: ifAbsent:
Heap removeFirst
Heap setCollection: tally:
IdentitySet scanFor:
InstructionStream interpretExtension: in: for:
InstructionStream interpretNextInstructionFor:
InstructionStream interpretNextInstructionFor:
Integer readFrom: base:
Integer readFrom:
Integer *
Integer ++
Integer --
Integer asCharacter
Integer asInteger
Integer copyto:
Integer denominator
Integer digitCompare:
Integer digitDiv: neg:
Integer digitMultiply: neg:
Integer digitSubtract:
OSPlatform class»determineActivePlatformStartingAt:
OSPlatform class»initialize
OSPlatform class»isMacOS
OSPlatform class»platformName
OSPlatform class»shutDown:
OSPlatform class»startUp:ISO88597
OSPlatform class»version
OSPlatform»shutDown:
OSPlatform»startUp:
Object class»flushDependents
Object class»flushEvents
Object class»initializeDependentsFields
Object class»initialize
Object class»newFrom:
Object»=
Object»actAsExecutor
Object»as:
Object»asSetElement
Object»asString
Object»assert:
Object»at:put:
Object»at:
Object»basicAt:put:
Object»basicAt:
Object»basicSize
Object»breakDependents
Object»class
Object»copyFrom:
Object»copySameFrom:
Object»copy
Object»enclosedSetElement
Object»encodeOn:
Object»executor
Object»greaseString
Object»hash
Object»instVarAt:put:
Object»instVarAt:
Object»instVarNamed:put:
Object»isArray
Object»isCharacter
Object»isInteger
Object»isKindOf:
Object»isLiteral
Object»isMemberOf:
Object»isSelfEvaluating
Object»isString
Object»isSymbol
Object»myDependents:
Object»notNil
Object»perform:with:
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 » waitForDataFor:
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 » new:
String » isEqual:
String » isString
String » isWideString
String » match:
String » putOn:
String » renderOn:
String » startingAt:
String » asString
String » sameAs:
String » asSymbol
String class » asDate
String class » asDateAndTime
String class » asNumber
String class » asScaledInteger
String class » asString
String class » asZnMimeType
String class » asZnUrl
String class » compare:caseSensitive:
String class » compare:with:collated:
String class » convertFromWithConverter:
String class » crlf
String class » dateAndTimeFromSeconds:
String class » dateAndTimeNow
String class » asTime
String class » asZnMimeType
String class » asZnUrl
TextConverter class » allEncodingNames
TextConverter class » forEncoding:
TextConverter class » latin1Encodings
TextConverter class » latin1MapGRCodec
TextConverter class » latin1MapGRCodec
TextConverter class » latin1MapGRCodec
Time class » hour
Time class » hour24
Time class » minute
Time class » millisecondClockValue
Time class » milliSecondsClock:
Time class » seconds
Time class » totalSeconds
Time class » whenClockTicks
Time class » second
Time class » second WhenClockTicks
Time class » seconds
Time class » seconds Since:
Time class » secondsSince:
Time class » seconds:since:
Time class » second:
Time class » seconds:
Time class » nanoSeconds:
Time class » nanoSecond:
Time class » nth
Time class » secondClock:
Time class » nanosecond:
Time class » nanoseconds:
Time class » milliseconds:
Time class » millisecond:
Time class » milliseconds:
Time class » milliseconds:
Time class » millisecond:
Time class » millisecond:
Time class » millisecond:
Time class » millisecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
Time class » nanosecond:
WAResponse status
WAResponseGenerator class:on:ShiftJIS
WAResponseGenerator class:setOn:
WAResponseGenerator requestContext
WAResponseGenerator request
WAResponseGenerator response
WAResponseGenerator respond
WAResponseGenerator response
WARoot class:context
WARoot setContext
WAScriptGenerator initialize
WAScriptGenerator load Scripts
WAScriptGenerator write Load Scripts On:
WAScriptGenerator write Script Tag: on;
WAServerAdaptor class: default Small
WAServerAdaptor class: manager:
WAServerAdaptor class: new
WAServerAdaptor class: port:
WAServerAdaptor class: start On:
WAServerAdaptor codec
WAServerAdaptor context For:
WAServerAdaptor default Port
WAServerAdaptor default Request Handler
WAServerAdaptor handle:
WAServerAdaptor handlePadding:
WAServerAdaptor handleRequest:
WAServerAdaptor initializeWith Manager:
WAServerAdaptor initialize
WAServerAdaptor manager
WAServerAdaptor port:
WAServerAdaptor port
WAServerManager class: default
WAServerManager class: initialize
WAServerManager class: shut Down
WAServerManager class: start Up
WAServerManager create Adaptors
WAServerManager can Start:
WAServerManager register:
WAServerManager start:
WASession action Field
WASession action Url For Continuation:
WASession action Url For Key:
WASession application
WASession clear Jump To
WASession create Cache
WASession status: message:
WAResponse status: