



Understanding Pharo's global state to move programs through time and space

Guillermo Polito, Noury Bouraqadi, Stéphane Ducasse, Luc Fabresse

► To cite this version:

Guillermo Polito, Noury Bouraqadi, Stéphane Ducasse, Luc Fabresse. Understanding Pharo's global state to move programs through time and space. International Workshop on Smalltalk Technologies, Aug 2014, Edinburgh, United Kingdom. hal-01070964

HAL Id: hal-01070964

<https://hal.archives-ouvertes.fr/hal-01070964>

Submitted on 2 Oct 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.

Understanding Pharo's global state to move programs through time and space

Guillermo Polito, Noury Bouraqadi, Stéphane Ducasse, Luc Fabresse

Mines-Telecom Institute, Mines Douai
RMOD INRIA Lille Nord Europe

Abstract

Code mobility is a mechanism that allows the migration of running programs between different environments. Such migration includes amongst others the migration of application data and resources. Application's data is usually composed by elements of different nature: from printers and files, to framework and domain objects. This application data will be transported along with the code of its program in space (when serialized and deployed in another environment) or time (when a new session is started in a different point of time). The main problem when moving around code resides, in our understanding, to *global* state. While unreferenced leaf objects are garbage collected, those referenced (transitively) by some global object will remain alive.

In order to support code mobility in time and space, we need to understand how global application data is used. With this purpose, we study and classify Pharo's global state. This classification uncovers some common patterns and provides a first insight on how global state should be managed, specially in code mobility scenarios. As a minor contribution, we also discuss solutions to each of the found categories.

1. Introduction

Code mobility is a mechanism that allows the migration of programs between different environments. It provides support for *e.g.*, load balancing, adjusting an application's resources dynamically and functionality customization. Fuggetta et al. define informally code mobility as the capability to re-bind a piece of code with the location it is running [?].

Such rebinding may consist, depending on the style of mobility, in the mobility of execution state, application data and resources, or both of them. Execution state mobility is the ability to suspend the actual execution of a program and transfer its internal execution information (*e.g.*, code, execution stacks, instruction pointers) to some other environment. Data mobility is the ability to transfer the application's data (*e.g.*, objects, database connections, files) between different environments.

Application data is usually composed by elements of different nature. Files are used for configuration and logging. Network connections such as sockets are used to communicate with remote systems. External libraries provide with code reuse. We can also find objects local to the application, of two different categories: domain objects modeling the application's specific concerns and application objects modeling those concerns that are cross-cutting between applications.

In our experience manipulating the language kernel of Pharo, we identified several cases where data mobility presents some issues. We can generalize those issues as mobility either *in time* (*i.e.*, creating or recreating a program), or *in space* (*i.e.*, moving a program between different environments):

Transporting code in space. When moving a program from one environment to another one, some of its state becomes invalid. For example, files existing in one machine will not exist in some other. Because of this, the migration mechanism should be aware of the state it migrates, to either reinitialize it, re-bind it in the new environment, or by keep it with its same value [?].

Transporting code in time. Image-based systems allow one to persist the state of a program to restart it at some other point of time from the last check-point, introducing the idea of *program sessions*: every time the system is restarted, a new *session* is started. These programming sessions introduce the concern of *session specific state* *i.e.*, state that is only valid during a programming ses-

sion. The mechanism in charge of stopping and restarting the image has to recognize the session specific state to re-initialize or rebind it every time a new session is started.

Creating for the first time. The initial creation of the system is a combination of transporting the program in space and time, since the issues of both appear in it. When creating or recreating the language kernel from scratch, for example during a bootstrap process [?], we must deal with its initialization. All the initial objects must be created, and their state is initialized by either binding it for the first time to some resource or assigning it some value. This state should be initialized in a proper order.

One of the main problems when moving code around resides in the existence of *global* state. While unreferenced leaf objects are garbage collected, those referenced (transitively) by some global object will remain alive. Because of this, we focus our attention on global state (cf. Section 2). Migrating global state in the cases described above in a generic way shows itself challenging (cf. Section 3). Global state is used for many different and unrelated purposes in the Pharo base libraries *e.g.*, from caches to constants values. Also, the intention of such usage is not explicit in the source code: its identification requires the developer to read the complete implementation.

In this paper we present an empirical study on the global state of Pharo base libraries. Our contribution is twofold:

- We present a classification of the usage of global state, identifying patterns built with global state constructs in Pharo (cf. Section 4).
- We discuss our findings and solutions to the issues we found. Our main goal with this is to make explicit those patterns. In such a way, client libraries and frameworks in charge of program migration can be simplified. (cf. Section 6 and Section 5).

2. Background

Global state is a simple and handy mechanism to share state between different objects. It is also a simple persistency mechanism: state hold by it will persist as long as the program is alive and running. Additionally, in image-based systems as Pharo it will remain alive through different program executions because the image persists its state taking as root the global objects. In this paper we put focus on global state because of this persistency property.

Global state is indeed not a bad mechanism per se, and is often used in applications to implement globally needed concerns. For example, Pharo implements through it a global process scheduler and the system dictionary holding all classes. However, its usage is discouraged in general terms because it introduces hidden dependencies in the software it is used.

2.1 Global State in Pharo

Global state in Pharo can be expressed in many forms with many constructs of the language. In this paper we will focus on the elements we present following. Note that equivalent language constructs can be found in other languages such as Java (for example, with static variables).

Global Variables. Global variables are variables that share their values to all objects in the system. A global variable can be accessed from any method, from any object. In Pharo, these kind of variables are stored in the global SystemDictionary object. Global variables may reference either (a) global instances such as Processor or Smalltalk, either (b) Classes and Traits.

Class Variables. Class variables are variables that belong to a class. These variables can be accessed by both classes and instances from the hierarchy below its owner class. Their value is shared between all the objects that can access them. In Pharo, these kind of variables are stored in a Dictionary object in its owner class.

Class Instance Variables. Class instance variables are instance variables of the classes. Their value is not directly accessible from subclasses and subinstances of the class. However, they are often made globally accessible with accessors.

Shared Pools. Shared pools are sets of class variables shared amongst many classes. Their values are accessible to all classes (and their instances) that import the shared pool. In Pharo, Shared Pools as implemented as classes treated specially by the compiler at binding time.

Method Literals. Each method contains a collection of those literal objects used in in *e.g.*, strings, literal arrays or numbers. As classes are globally accessible, their methods are too, and so their literals.

2.2 About the State in Image-Based Systems

Pharo, as a Smalltalk inspired language, is an image-based language such as Lisp. Image-based languages present the following two main properties: direct object manipulation and persistence. Direct object manipulation provides with instant feedback during development and a flexible way to understand the state of applications. Persistence allows one to store those changes made by direct object manipulation without the need of recreating the system every time it is started. Indeed, an image-based language can be persisted and restarted later on, possibly in another machine. We refer as a *session* to the time elapsed between the startup of an image and its shutdown.

These programming sessions have *session specific state*, *i.e.*, state that is valid only within a session. For example, we can name as such file and socket descriptors, handles to external libraries, operating system information, and time and date information. These kind of objects become invalid

when their session is finished. Using them in an invalid state may lead to unexpected behavior, exceptions and virtual machine crashes. The language runtime must ensure that this state is correctly handled on session startup and shutdown: *e.g.*, reinitialize it or discard it.

3. Motivation

In this section we show why understanding and making explicit the usage of global state is important. We introduce first an example based on two Pharo's cache implementations, and their problems. Then, we explore three different situations in which those problems are made more evident.

3.1 Problems on Global State Usage: an Example

To exemplify the problems on global state usage, we present here two different global cache implementations we find in Pharo 3.0. First, in Figure 1, we present a simplified version of the AST cache. Second, in Figure 2, we find an extract of the HelpIcon class, with the code related to an icon cache. By looking at these two ad-hoc implementations of caches, we identify the following issues:

Incompleteness. Both cache implementations were written to solve only particular issues. The AST cache presents weak references as it inherits from the WeakIdentityKeyDictionary class and also presents methods to be flushed. The icon cache does not present code for any of those features. None of them cover some concerns a cache may want to address such as specifying a maximum amount of elements or a recycling strategy (LRU, FIFO, etc.).

Non-Explicitness. In order to identify the examples as caches we need to read their code: the names of the classes and variables gives us an idea of its responsibility as caches. The default method in the AST cache hints us about having found also a singleton. This problem uncovers the existence of **hidden information** in the system. One cannot query the system to, for example, obtain a list of the existing caches in order to flush them, or make a report on their memory usage.

3.2 Creating Programs from Scratch: Bootstrapping

While bootstrapping Pharo [?], we must initialize the image's global state. We observed the need for an order in this initialization, showing off a hidden coupling between code pieces. For example, some global tables must be initialized before initializing the classes state, which in turn must be initialized before the rest of the language kernel (*i.e.*, the startup and shutdown lists, the main processes, etc.).

Since the global state language constructs are used for different concerns **implicitly**, it is difficult to discern whether they are responsibility of the language kernel, of basic libraries such as Collections, or other not-basic ones such as Networking. This makes the bootstrap process difficult to maintain. A lot of ad-hoc code should be written to handle

```

1 WeakIdentityKeyDictionary subclass: #ASTCache
2   classVariableNames: 'Default'.
3
4 ASTCache>>at: aCompiledMethod
5   ^ self
6     at: aCompiledMethod
7     ifAbsentPut: [ self newASTFor: aCompiledMethod ].
8
9 ASTCache>>newASTFor: aMethod
10  "creation of the AST..."
11
12 ASTCache>>reset
13  self removeAll.
14
15 ASTCache class>>default
16  ^ Default ifNil: [ Default := self new ].
17
18 ASTCache class>>shutDown
19  self default reset.
```

Figure 1. Simplified code of Pharo's AST cache.

```

1 Object subclass: #HelpIcons
2   classVariableNames: 'Icons'.
3
4 HelpIcons>>icons
5   ^ Icons ifNil: [Icons := Dictionary new]
6
7 HelpIcons>>iconNamed: aSymbol
8   ^ self icons at: aSymbol ifAbsentPut: [self perform: aSymbol]
9
10 HelpIcons>>refreshIcon
11  ^ "creates a new icon object"
```

Figure 2. Code of Pharo's Help Icon class with an icon cache.

the dependencies between the global state in Pharo's language kernel.

3.3 Transporting Programs in Time: Session Awareness

Image-based systems introduce the concern of *session specific state*. State holding references to for example, files, caches, or platform specific information, may become invalid when a new session is started in a different moment. Pharo presents a startup and a shutdown mechanism to support this. The language runtime raises events on its startup and shutdown. Classes subscribed to such events are notified and will execute some code according to the event. The handler of these events is responsibility of the class developer. This mechanism hides information in two different levels:

Dependencies between classes. The subscribed classes receive the startup and shutdown events in an explicitly defined order. This order is present in a list which is defined by the developers. This list express a dependency between classes *e.g.*, some classes must receive the startup event before others to satisfy its invariants. However, this

list does not actually express the reason of this dependency *i.e.*, which is the state or invariants that should be guaranteed before each class receives the proper event.

Semantics of class state. The startup and shutdown event handlers, which are in charge of the clean-up and reinitialization of some the global state, are written in an imperative fashion. This imperative fashion hides the semantics and invariants of this state.

This hidden information makes difficult to change the startup and shutdown mechanism. Some questions appear when doing so: Can we remove some class from these lists? Can we alter the order without changing the behavior? When we register a new class, in which position should we put it?

3.4 Transporting Programs in Space: Serialization

Migrating objects, and specially code (classes and methods), from one image to another requires in general customizations for the global state it carries and references. References to external classes and global variables may not be serialized but just re-bound in the new environment. Class variables containing constant value objects may be transported with the program. Session specific state should be re-initialized, as program migration implies session change also.

A migration mechanism needs information about the semantics of the state in migration, so it knows whether it should reinitialize it, re-bind it, or keep it as it is. As this information is **not usually explicitly available** in the program under migration, the developer must add it in the form of extensions or descriptions, external to the program. For example, the serialization library Fuel [?] presents special clusters to handle and customize the serialization of global variables and class variables. The user must customize these clusters externally.

4. Classification of Pharo's Global State

4.1 Classification Methodology

Our universe of study is the latest release of Pharo, Pharo 3.0. We selected as individuals to study all those usages of global state language constructs as we presented it in Section 2 *i.e.*, class variables and class instance variables, shared pools and global variables. For simplicity, we excluded from our analysis the classes referenced by global variables. We also excluded method literals because analyzing them would mean to read every single method in the language kernel.

The global state in Pharo is present mostly in ad-hoc implementations, making difficult the usage of automated methods for its classification. Since the goal of this paper is not to obtain an automatic classification, we built our classification using purely empirical observation: reading the code. We took each of the selected individuals, read all the code related to it and made a qualitative evaluation of it. We put special emphasis on the side-effects on such individuals, which showed useful to recognize the individual's semantics in the program.

As a result, we distinguished some patterns of usage, which lead us to the categories in Section 4.2. Note that the individuals under study can fall into more than of these categories *e.g.*, a cache made globally as a singleton. Also, to avoid noise we excluded from the classification those individuals whose role in the source code was very specific, thus they did not conform a representative category.

4.2 Categories

Constants. Constants are values that are initialized once and never updated. Pharo has no construct to express constant values. Thus, they are expressed using the other available constructs. This means that the semantics of constants must be ensured explicitly in the code or they are not ensured at all.

Settings and Configurable Default Values. Settings and configurable default values provide a single point to configure and share values amongst several instances. They are publicly accessible so they can be modified and customized by developers. Pharo uses settings to store for example maximum size of UI widgets, code completion configurations and network configurations.

Singletons. Singletons are well known objects globally accessible in the system [? ?]. They are used to provide a single access point to some shared state or behavior. Pharo presents several different singleton implementations: global leaf objects (not classes nor traits) such as *e.g.*, the Processor or the Transcript, leaf objects stored in class variables or class instance variables often accessible through the `uniqueInstance` message, and some classes which are indeed used as singletons.

Caches. A cache is a buffer that stores duplicated information to reduce the consumption of resources such as CPU or memory. Caches store usually up to a maximum amount of elements, discarding old ones following a given strategy *e.g.*, First In First Out (FIFO) and Least Recently Used (LRU). Pharo presents several caches which store for example images, fonts and package metadata.

Registries. A registry stores a list of possible service providers and resolves which one of them is the appropriate to handle a task. They are usually used as a factory, to decouple the users of a service from a particular implementation. For example, a compiler registry may store all the compiler implementations available and provide a default one. A registry allows users to subscribe and unsubscribe services into it. For example, when a notification has to be shown to the end user, the UIManager registry decides how to show it according to its registered providers: either by using the standard output or the graphical user interface. Pharo uses registries to manage different kind of concerns such as the compiler suite, the fonts or the UI interactions.

Session Specific State. Session specific state is the global state that is tied to a particular session *e.g.*, information gathered from the current platform, file handles and library handles. This state should be reinitialized or reset when a new session is started either in a new machine or a different one, to avoid misbehaviors and unexpected errors.

Process Controllers. Process controllers manage the life cycle of well known processes such as the idle process, the user interface (UI) process or the low space watcher process. They control how and when these well known processes are started, terminated, suspended and resumed.

Finalizables. Resources external to the language, such as files, sockets, or handles to external libraries, must be finalized accordingly when they are garbage collected or new session are started. For such a task, the classes of those objects implement a finalization mechanism to be aware of garbage collections and handle such situations.

Graphical Resources. Graphical resources are objects such as images, fonts, icons or bitmaps. These resources are embedded in the system using the global state constructs. As such, there is no general solution to discard them or reload them.

4.3 Results and Discussion of Impact

Table 1 lists the results of applying each of our categories to our set of individuals under study: how many of them apply to each category. The details of such a classification can be found in the Appendix A.

These results present some particularities we should take into account before doing a deep analysis. First, the number of detected graphical resources does not really represent the reality. A lot of graphical resources are represented as byte arrays in method literals (which we did not measure because of its complexity). With respect to the numbers in our results, we can argue that they give us an idea of the impact produced by each category *i.e.*, code-migration libraries have to potentially handle each appearance of these patterns in an ad-hoc fashion, since they are not explicit in the source code. For example, if we would decide that on serialization all caches should be flushed, we must add custom code to handle each of the 43 caches.

5. Discussion: a need for Reification

5.1 Concepts to Reify in Pharo.

Bouraqadi et al. [?] presented already the need for the reification of resources used in a mobile code. The reification of resources provide support for an open architecture and facilitates the task of object migration. They also make explicit the concepts that are part of the program, providing with information the system can benefit from. We identify in

Category	Amount satifying
Constants	1722
Settings and Configurable Default Values	236
Singletons	65
Graphical Resources*	47
Caches	43
Registries	31
Session Specific	27
Process Controllers	11
Finalizables	6

Table 1. Amount of individuals classified under each of the identified categories.

particular the need for reification of the following elements part of our categories:

Processes. Pharo processes, although they are already objects, are managed from other objects. Process specific state is controlled by objects other than the process itself, breaking encapsulation. As such, the life-cycle of processes are tied to those objects that create them or keep their state. A first class representation of processes, on the other hand, will encapsulate the process specific state, avoid conflicts on its access, and provide a common interface for their manipulation.

Finalizables. First class finalizable resources provide a framework supporting uniform finalization and resource deallocation.

Caches. First class caches provide a uniform and complete implementation of caches libraries can rely upon. Additionally they will enable the system with introspection and self-modification of such caches.

Variables. First class variables, namely slots, were sketched by Verwaest et al. [?] and a first version introduced into Pharo 3.0. Slots introduce the ability to refine instance variables, give them specific behavior and annotate them with meta-information. Specialized slots can be used to implement *e.g.*, session specific state, constant values or settings.

5.2 Using explicit metaclasses.

Finding all singletons installed in the system could be easily achieved through the usage of explicit metaclasses [?] or traits [?]. Explicit metaclasses and traits allow the sharing of behavior between classes, and thus, they eliminate the need for ad-hoc implementations of *e.g.*, singletons. Additionally, reifying the singleton abstraction in the language, provides with the ability to query and act upon the installed singletons. Implementing them with traits, however, presents as main limitation that the current trait implementation in Pharo is stateless. Thus, it does not allow to express class variables to hold the singleton instance.

6. Discussion: Moving responsibilities to the language runtime

Within our classification, we understand there are some concerns that should be moved under the umbrella of the language or its runtime system. The language may provide its own abstractions for recurrent problems such as caching or registering services. This will provide with the proper and needed meta-information to handle services. Additionally, providing end users with correct and complete implementations will avoid the ad-hoc implementations with repeated logic.

6.1 Resource manager

As we noted in the results, graphical resources such as images, icons and fonts are present as globally accessible resources in Pharo. We can add also that Pharo's memory is occupied in great percentage by instantiated bitmaps¹ [?]. There is not, however, a possibility to inspect all available resources, understand their origin (the package, class and method that defines them), or recreate them from files. This poses the need for a resource manager.

A sketch implementation of such a resource manager was implemented as a in-memory file system. In such a prototype, each Pharo package contains an associated file system that stores resources of that package. Images, icons, configuration files, and other files are stored in this file system. Package resources can be accessed from within and outside the package in an structured way, and serialized along with its package.

6.2 Session manager

How session specific state is handled nowadays denotes the need for a session manager. Currently, in the presence of session specific state, the class that stores it has to be subscribed to the startup and shutdown events of the runtime system. These two events are used to reset and initialize the class state when a new session is started.

We sketched a session manager to ease the management of session specific state. First class instance variables (Slots) describe declaratively their initialization when a new session is started. Then, during the startup of a new session, the session manager will reinitialize each of these slots using their description. This session manager encapsulates the need for the startup and shutdown lists, and removes such responsibility from the developer.

7. Related Work

Fuggetta et al. [?] present also a classification of the state of mobile systems, but using as criteria the strategy used for migration. As such, their classification is orthogonal and complementary to ours. They present two properties to characterize the data to migrate

Transferrable. A transferrable element is the one that can be physically migrated *e.g.*, a file. Oppositely, a non transferrable one is the one that cannot be migrated, *e.g.*, as a printer.

Desirability to transfer it. An application can mark some data as *fixed* or *free* according to its needs. Fixed data is associated permanently with its original environment, while free data migration is allowed.

and three ways to bind an application to a given resource

By identifier. Resources binded by identifier are tied with a particular instance of a resource *e.g.*, a socket. When a program is migrated, all its resources binder by identifier are kept in their original environment. A network communication is enforced between them.

By value. Resources binded by value are interested in the value of a resource and not in their identity, *e.g.*, the contents of a file. These kind of resources can be copied along with the program upon migration.

By type. Resources binded by type are intended to provide some kind of service despite their value or identity *e.g.*, a display. These kind of resources are rebinded to local resources of the same type after migration.

Ungar et al. implemented a transporter for the Self programming language [?]. This transporter had to deal with many of the difficulties we presented above, in particular the lack of explicit usage information. They provided a generic solution to the problem: let the developer annotate the objects' slots to guarantee the desired state of the program upon a migration. However, a question remained: How should developers annotate the slots? To answer this question, they provided with a series of properties that must help in such analysis.

Does identity matter? The developer has to identify those objects whose identity matters, and those whose it doesn't. When identity matters, the transporter must ensure that references to the same object are kept the same after migration. When it does not, the transporter can simply duplicate the object.

An initial value must always be enforced? Some objects must be reinitialized every time they are migrated. This is for example the case of caches.

An object must be written in an abstract or concrete way? Some objects can be rebuilt as the result of an expression, while some others must be built by concretely enumerating its slots.

8. Conclusion and Future Work

In this paper we studied the usage of global state in Pharo. The study of global state is interesting since references kept from global state are persisted in image-based systems. Global state is also a concern in when working in

¹ 24.50% according to our measures in latest Pharo version

code mobility because resources globally available must be reinitialized or rebinded when code is migrated.

We present a classification of Pharo's global state based on its usage, and found many patterns that are recurrent in the kernel of the language, though not explicit in the code. We discuss how to make explicit these patterns so the language kernel can benefit from it, either by reifying them or moving some responsibilities to the language kernel.

This work is a first step to prepare Pharo to the mobile code world. To be able to transport Pharo programs either in time or space, the abstractions we found should be made explicit in the language, and so, libraries and frameworks

can take advantage of them. As future work we also consider that the discussed sketches have to be iterated and developed further.

Acknowledgements

This work was supported by Ministry of Higher Education and Research, Nord-Pas de Calais Regional Council, FEDER through the 'Contrat de Projets Etat Region (CPER) 2007-2013', the Cutter ANR project, ANR-10-BLAN-0219 and the MEALS Marie Curie Actions program FP7-PEOPLE-2011- IRSES MEALS.

A. Appendix: Classification

A.1 Finalizables

FileHandle -> #Registry
FT2Handle -> #Registry
Socket -> #Registry
StandardFileStream -> #Registry
WeakRegistry -> #Default

A.2 Process Controllers

CPUWatcher -> #CurrentCPUWatcher
Delay -> #TimerEventLoop
MessageTally -> #Timer
MorphicUIManager -> #UIProcess
ProcessBrowser -> #SuspendedProcesses
ProcessBrowser -> #WellKnownProcesses
ProcessorScheduler -> #BackgroundProcess
SmalltalkImage -> #LowSpaceProcess
UpdateStreamer -> #UpdateDownloader
WeakArray -> #FinalizationProcess

A.3 Registries

Beeper -> #default
ChangeSet -> #AllChangeSets
ChangeSet -> #current
EncodedCharSet -> #EncodedCharSets
ExternalDropHandler -> #DefaultHandler
ExternalDropHandler -> #RegisteredHandlers
FileServices -> #FileReaderRegistry
FreeTypeFontProvider -> #current
FreeTypeGlyphRenderer -> #current
HelpBrowser -> #DefaultHelpBrowser
LanguageEnvironment -> #ClipboardInterpreterClass
LanguageEnvironment -> #Current
LanguageEnvironment -> #FileNameConverter
LanguageEnvironment -> #InputInterpreterClass
LanguageEnvironment -> #KnownEnvironments
LanguageEnvironment -> #SystemConverter
Locale -> #KnownLocales
MCPackageManager -> #registry
MCServerRegistry -> #registry
MetacelloProjectRegistration -> #registry
Nautilus -> #PluginClasses
PluggableTextMorph -> #StylingClass
RBProgramNode -> #FormatterClass
RGFactory -> #CurrentFactories
SmalltalkImage -> #CompilerClass
SmalltalkImage -> #Tools
SoundSystem -> #Current
TestResource -> #current
UIManager -> #Default
UITheme -> #Current
ZnServer -> #ManagedServers
ZnSingleThreadedServer -> #Default

A.4 Caches

ASTCache -> #default
AbstractMethodWidget -> #MethodsIconsCache

AbstractNautilusUI -> #ClassesIconsCache
AbstractNautilusUI -> #GroupsIconsCache
AbstractNautilusUI -> #PackagesIconsCache
BitBlt -> #CachedFontColorMaps
BitBlt -> #ColorConvertingMaps
CairoBackendCache -> #soleInstance
Color -> #CachedColormaps
Color -> #MaskingMap
FreeTypeCache -> #current
GLMUIThemeExtraIcons -> #icons
GradientFillStyle -> #PixelRampCache
HelpIcons -> #Icons
KomitClass -> #classes
KomitMethod -> #methods
KomitPackage -> #packages
KomitRemote -> #icon
Komitter -> #lastMessage
LoadingMorphState -> #image
LogicalFont -> #all
MCDefinition -> #Instances
MCGitHubRepository -> #DownloadCache
MCMethodDefinition -> #Definitions
MCSaveVersionDialog -> #PreviousMessages
NECSymbols -> #cachedSymbols
RPackageSet -> #cachePackages
ScrollBar -> #ArrowImagesCache
ScrollBar -> #BoxesImagesCache
SettingDeclaration -> #ValueListCache
SingleCodeCriticResultList -> #icons
SugsSuggestionFactory -> #collectorForAll
SugsSuggestionFactory -> #collectorForAssignment
SugsSuggestionFactory -> #collectorForClassVariable
SugsSuggestionFactory -> #collectorForClass
SugsSuggestionFactory -> #collectorForInstancesVariable
SugsSuggestionFactory -> #collectorForLiteral
SugsSuggestionFactory -> #collectorForMessage
SugsSuggestionFactory -> #collectorForMethod
SugsSuggestionFactory -> #collectorForSourceCode
SugsSuggestionFactory -> #collectorForTemporaryVariable
SugsSuggestionFactory -> #collectorForUndeclaredVariable

A.5 Graphical Resources

AbstractMethodWidget -> #MethodsIconsCache
AbstractNautilusUI -> #ClassesIconsCache
AbstractNautilusUI -> #GroupsIconsCache
AbstractNautilusUI -> #PackagesIconsCache
Cursor -> #BlankCursor
Cursor -> #BottomLeftCursor
Cursor -> #BottomRightCursor
Cursor -> #CornerCursor
Cursor -> #CrossHairCursor
Cursor -> #CurrentCursor
Cursor -> #DownCursor
Cursor -> #MarkerCursor
Cursor -> #MenuCursor
Cursor -> #MoveCursor
Cursor -> #NormalCursor

Cursor -> #OriginCursor
 Cursor -> #OverEditableText
 Cursor -> #ReadCursor
 Cursor -> #ResizeLeftCursor
 Cursor -> #ResizeTopCursor
 Cursor -> #ResizeTopLeftCursor
 Cursor -> #ResizeTopRightCursor
 Cursor -> #RightArrowCursor
 Cursor -> #SquareCursor
 Cursor -> #TargetCursor
 Cursor -> #TopLeftCursor
 Cursor -> #TopRightCursor
 Cursor -> #UpCursor
 Cursor -> #WaitCursor
 Cursor -> #WebLinkCursor
 Cursor -> #WriteCursor
 Cursor -> #XeqCursor
 FreeTypeCache -> #current
 FreeTypeSettings -> #current
 GLMUIThemeExtraIcons -> #icons
 HelpIcons -> #Icons
 IconicButton -> #DefaultGraphics
 ImageMorph -> #DefaultForm
 LogicalFontManager -> #current
 RemotesManager -> #addRemoteIcon
 RemotesManager -> #editRemoteIcon
 RemotesManager -> #removeRemoteIcon
 ScrollBar -> #ArrowImagesCache
 ScrollBar -> #BoxesImagesCache
 SingleCodeCriticResultList -> #icons
 Transcriber -> #Icon
 TransferMorph -> #CopyPlusIcon

A.6 Session Specific State

MCGitHubRepository -> #DownloadCache
 MCCacheRepository -> #default
 DiskStore -> #CurrentFS
 NOCCompletionTable -> #table
 NOCCompletionTable -> #classTable
 Locale -> #Current
 Locale -> #CurrentPlatform
 DateAndTime -> #LocalTimeZone
 FT2Handle -> #Session
 FileLocator -> #Resolver
 FileStream -> #Stdin
 FileStream -> #Stdout
 FileStream -> #TheStdioHandles
 FileStream -> #StdioFiles
 FileStream -> #Stderr
 LanguageEnvironment -> #SystemConverter
 LanguageEnvironment -> #FileNameConverter
 UUIDGenerator -> #Default
 VirtualMachine -> #WordSize
 WeakFinalizationList -> #HasNewFinalization
 AthensCairoSurface -> #uniqueSession
 AthensCairoSurface -> #dispatch
 AthensCairoSurface -> #dispatchStruct
 CairoLibraryLoader -> #session
 CairoLibraryLoader -> #libHandle

Session -> #current
 MultiByteFileStream -> #LineEndDefault

A.7 Singletons

ASTCache -> #default
 ActiveEvent -> #ActiveEvent
 ActiveHand -> #ActiveHand
 ActiveWorld -> #ActiveWorld
 Author -> #uniqueInstance
 BorderStyle -> #Default
 CPUWatcher -> #CurrentCPUWatcher
 CairoBackendCache -> #soleInstance
 ChangesLog -> #DefaultInstance
 Clipboard -> #Default
 CommandLineArguments -> #singleton
 CriticWorkingConfiguration -> #Current
 Display -> #Display
 EditorFindReplaceDialogWindow -> #Singleton
 EmptyLayout -> #instance
 FreeTypeCache -> #current
 FreeTypeSettings -> #current
 IdentityTransform -> #Default
 InputEventFetcher -> #Default
 KMBuffer -> #uniqueInstance
 KMPragmaKeymapBuilder -> #UniqueInstance
 KMRepository -> #Singleton
 KomitterManager -> #instance
 LayoutEmptyScope -> #instance
 LogicalFontManager -> #current
 MBConfigurationRoot -> #Current
 MCFileTreeFileUtils -> #Current
 MCRepositoryGroup -> #default
 MCServerRegistry -> #uniqueInstance
 MetacelloPlatform -> #Current
 NBExternalResourceManager -> #soleInstance
 NECController -> #uniqueInstance
 NNavNavigation -> #Instance
 NNavNavigation -> #Instance
 NativeBoost -> #Current
 NautilusMonticello -> #Default
 OSPlatform -> #Current
 PackageOrganizer -> #default
 PharoFilesOpener -> #Default
 PharoTutorial -> #Instance
 ProcessSpecificVariable -> #soleInstance
 Processor -> #Processor
 RBRefactoringManager -> #Instance
 RBRefactoryChangeManager -> #Instance
 RPackageOrganizer -> #default
 RecentMessageList -> #UniqueInstance
 Sensor -> #Sensor
 SharedValueHolder -> #instance
 Smalltalk -> #Smalltalk
 SoundTheme -> #Current
 SourceFiles -> #SourceFiles
 Spotlight -> #Current
 StartupPreferencesLoader -> #UniqueInstance
 SystemAnnouncer -> #announcer

SystemOrganization -> #SystemOrganization
 SystemProgressMorph -> #UniqueInstance
 SystemVersion -> #Current
 Transcript -> #Transcript
 UUIDGenerator -> #Default
 Undeclared -> #Undeclared
 UserManager -> #default
 VTermOutputDriver -> #stderrTerminalInstance
 VTermOutputDriver -> #stdoutTerminalInstance
 World -> #World
 ZnNetworkingUtils -> #Default

A.8 Settings and Configurable Default Values

AbstractNautilusUI -> #NextFocusKey
 AbstractNautilusUI -> #PreviousFocusKey
 AlphaImageMorph -> #DefaultImage
 BalloonMorph -> #BalloonFont
 CCompilationContext -> #WarningAllowed
 CPUWatcher -> #CpuWatcherEnabled
 ChangeSet -> #DefaultChangeSetDirectoryName
 ChangeSet -> #MustCheckForSlips
 CodeHolder -> #AnnotationRequests
 CodeHolder -> #BrowseWithPrettyPrint
 CodeHolder -> #DecorateBrowserButtons
 CodeHolder -> #DiffsInChangeList
 CodeHolder -> #DiffsWithPrettyPrint
 CodeHolder -> #OptionalButtons
 CodeHolder -> #ShowAnnotationPane
 CodeHolder -> #SmartUpdating
 CommandLineUIManager -> #SnapshotErrorImage
 DangerousClassNotifier -> #enabled
 Deprecation -> #RaiseWarning
 Deprecation -> #ShowWarning
 DialogItemsChooserUI -> #alreadySearchedSelectedItemList-
 MaxSize
 DialogItemsChooserUI -> #alreadySearchedUnselectedItemsList-
 MaxSize
 DisplayScreen -> #DeferringUpdates
 DisplayScreen -> #DisplayChangeSignature
 DisplayScreen -> #LastScreenModeSelected
 DisplayScreen -> #ScreenSave
 Editor -> #BlinkingCursor
 Editor -> #CmdKeysInText
 Editor -> #DumbbellCursor
 Editor -> #SkipOverMultipleSpaces
 EyeInspector -> #useAutoRefresh
 FLCompiledMethodCluster -> #transformationForSerializing
 FinderUI -> #Icon
 FinderUI -> #searchedTextListMaxSize
 Form -> #FloodFillTolerance
 FreeTypeSettings -> #UpdateFontsAtImageStartup
 FreeTypeSystemSettings -> #LoadFT2Library
 GrowlMorph -> #DefaultBackgroundColor
 GrowlMorph -> #Position
 HaloMorph -> #CurrentHaloSpecifications
 HaloMorph -> #HaloEnclosesFullBounds
 HaloMorph -> #HaloWithDebugHandle
 HaloMorph -> #ShowBoundsInHalo
 HandMorph -> #DoubleClickTime

HandMorph -> #NormalCursor
 HandMorph -> #ShowEvents
 HandMorph -> #UpperHandLimit
 Heap -> #sortBlock
 LongTestCase -> #RunLongTestCases
 MBInfo -> #ValidateAll
 MCDirectoryRepository -> #DefaultDirectoryName
 MCFileRepositoryInspector -> #Order
 MCFileTreeRepository -> #defaultPackageExtension
 MCFileTreeRepository -> #defaultPropertyFileExtension
 MCGitHubRepository -> #CacheDirectory
 MCMMethodDefinition -> #InitializerEnabled
 MCWorkingCopyBrowser -> #Order
 MCWorkingCopyBrowser -> #ShowOnlyRepositoriesFromWork-
 ingCopy
 MCWorkingCopyBrowser -> #repositorySearchMaxSize
 MCWorkingCopyBrowser -> #workingCopySearchMaxSize
 MessageDialogWindow -> #AutoAccept
 MessageTally -> #DefaultPollPeriod
 MetacelloCommonMCSpecLoader -> #RetryPackageResolution
 MetacelloScriptEngine -> #DefaultRepositoryDescription
 MetacelloScriptEngine -> #DefaultVersionString
 MonticelloRepositoryBrowser -> #Order
 Morph -> #CmdGesturesEnabled
 Morph -> #CycleHalosBothDirections
 Morph -> #DefaultYellowButtonMenuEnabled
 Morph -> #HalosEnabled
 MorphicModel -> #KeyboardFocusOnMouseDown
 MorphicModel -> #MouseOverForKeyboardFocus
 NECPreferences -> #backgroundColor
 NECPreferences -> #captureNavigationKeys
 NECPreferences -> #caseSensitive
 NECPreferences -> #enabled
 NECPreferences -> #expandPrefixes
 NECPreferences -> #popupAutomaticDelay
 NECPreferences -> #popupShowAutomatic
 NECPreferences -> #popupShowWithShortcut
 NECPreferences -> #smartCharactersMapping
 NECPreferences -> #smartCharactersWithDoubleSpace
 NECPreferences -> #smartCharactersWithSingleSpace
 NECPreferences -> #smartCharacters
 NECPreferences -> #spaceAfterCompletion
 NECPreferences -> #useEnterToAccept
 NNavNavigation -> #UseArrowShortcuts
 Nautilus -> #CommentPosition
 Nautilus -> #HistoryMaxSize
 Nautilus -> #OpenOnGroups
 Nautilus -> #ShowAnnotationPane
 Nautilus -> #ShowHierarchy
 Nautilus -> #SwitchClassesAndPackages
 Nautilus -> #WarningLimit
 Nautilus -> #emptyCommentWarning
 Nautilus -> #maxSize
 Nautilus -> #populateMethodList
 Nautilus -> #useOldStyleKeys
 NautilusRefactoring -> #PromptOnRefactoring
 NetNameResolver -> #DefaultHostName
 NetworkSystemSettings -> #BlabEmail
 NetworkSystemSettings -> #HTTPProxyExceptions

NetworkSystemSettings -> #HTTPProxyPort
 NetworkSystemSettings -> #HTTPProxyServer
 NetworkSystemSettings -> #ProxyPassword
 NetworkSystemSettings -> #ProxyUser
 NetworkSystemSettings -> #UseHTTPProxy
 NetworkSystemSettings -> #UseNetworkAuthentication
 ObjectExplorer -> #ShowIcons
 PSMCPatchMorph -> #UsedByDefault
 PackageTreeNautilus -> #ShowGroupsOnTop
 Paragraph -> #InsertionPointColor
 Path -> #absoluteWindowsPathRegex
 PluggableButtonMorph -> #UseGradientLook
 PluggableTextMorph -> #ShowTextEditingState
 PluggableTextMorph -> #StylingClass
 PluggableTextMorphWithLimits -> #DefaultWarningLimit
 PolygonMorph -> #CurvierByDefault
 PolymorphSystemSettings -> #DesktopColor
 PolymorphSystemSettings -> #DesktopColor
 PolymorphSystemSettings -> #DesktopGradientDirection
 PolymorphSystemSettings -> #DesktopGradientDirection
 PolymorphSystemSettings -> #DesktopGradientFillColor
 PolymorphSystemSettings -> #DesktopGradientFillColor
 PolymorphSystemSettings -> #DesktopGradientOrigin
 PolymorphSystemSettings -> #DesktopGradientOrigin
 PolymorphSystemSettings -> #DesktopImageFileName
 PolymorphSystemSettings -> #DesktopImageFileName
 PolymorphSystemSettings -> #DesktopLogoFileName
 PolymorphSystemSettings -> #DesktopLogoFileName
 PolymorphSystemSettings -> #DesktopLogo
 PolymorphSystemSettings -> #DesktopLogo
 PolymorphSystemSettings -> #ShowDesktopLogo
 PolymorphSystemSettings -> #ShowDesktopLogo
 PolymorphSystemSettings -> #ShowDesktopLogo
 PolymorphSystemSettings -> #UseDesktopGradientFill
 PolymorphSystemSettings -> #UseDesktopGradientFill
 PolymorphSystemSettings -> #usePolymorphDiffMorph
 PolymorphSystemSettings -> #usePolymorphDiffMorph
 PolymorphSystemSettings -> #usePolymorphDiffMorph
 ProgressBarMorph -> #DefaultHeight
 ProgressBarMorph -> #DefaultWidth
 ProportionalSplitterMorph -> #ShowHandles
 RBConfigurableFormatter -> #CascadedMessageInsideParentheses
 RBConfigurableFormatter -> #FormatCommentWithStatements
 RBConfigurableFormatter -> #IndentString
 RBConfigurableFormatter -> #IndentsForKeywords
 RBConfigurableFormatter -> #KeepBlockInMessage
 RBConfigurableFormatter -> #LineUpBlockBrackets
 RBConfigurableFormatter -> #MaxLineLength
 RBConfigurableFormatter -> #MethodSignatureOnMultipleLines
 RBConfigurableFormatter -> #MinimumNewLinesBetweenStatements
 RBConfigurableFormatter -> #MultiLineMessages
 RBConfigurableFormatter -> #NewLineAfterCascade
 RBConfigurableFormatter -> #NewLineBeforeFirstCascade
 RBConfigurableFormatter -> #NewLineBeforeFirstKeyword
 RBConfigurableFormatter -> #NewLinesAfterMethodComment
 RBConfigurableFormatter -> #NewLinesAfterMethodPattern
 RBConfigurableFormatter -> #NewLinesAfterTemporaries
 RBConfigurableFormatter -> #NumberOfArgumentsForMultiLine
 RBConfigurableFormatter -> #OneLineMessages
 RBConfigurableFormatter -> #PeriodsAtEndOfBlock
 RBConfigurableFormatter -> #PeriodsAtEndOfMethod
 RBConfigurableFormatter -> #RetainBlankLinesBetweenStatements
 RBConfigurableFormatter -> #StringFollowingReturn
 RBConfigurableFormatter -> #StringInsideBlocks
 RBConfigurableFormatter -> #StringInsideParentheses
 RBConfigurableFormatter -> #TraditionalBinaryPrecedence
 RBConfigurableFormatter -> #UseTraditionalBinaryPrecedenceForParentheses
 RBRefactoring -> #RefactoringOptions
 RBRefactoryChangeManager -> #UndoSize
 RealEstateAgent -> #StaggerOffset
 RealEstateAgent -> #StandardSize
 RealEstateAgent -> #UsedStrategy
 RecentMessageList -> #settingDropList
 SHPreferences -> #CustomStyleTable
 SHPreferences -> #Groups
 SHPreferences -> #enabled
 SHTextStyleerST80 -> #styleTable
 SHTextStyleerST80 -> #textAttributesByPixelHeight
 ScriptLoader -> #CheckImageSyncWithUpdate
 SettingBrowser -> #regexpSearch
 SettingBrowser -> #searchedTextList
 SimpleEditor -> #CmdActions
 SimpleEditor -> #ShiftCmdActions
 SmalltalkImage -> #ShouldDownloadSourcesFile
 SoundSystem -> #SoundEnabled
 SoundSystem -> #SoundQuickStart
 SoundTheme -> #UseThemeSounds
 SpecDebugger -> #AlwaysOpenFullDebugger
 SpecDebugger -> #ErrorRecursion
 SpecDebugger -> #FilterCommonMessageSends
 SpecDebugger -> #LogDebuggerStackToFile
 SpecDebugger -> #LogFileName
 SpecDebuggerStack -> #DoItFilterEnabled
 SpecDebuggerStack -> #FilterDictionary
 SpecDebuggerStack -> #KCFilterEnabled
 SpecDebuggerStack -> #NilSelectorsFilterEnabled
 StandardFonts -> #ButtonFont
 StandardFonts -> #CodeFont
 StandardFonts -> #HaloFont
 StandardFonts -> #ListFont
 StandardFonts -> #MenuFont
 StandardFonts -> #WindowTitleFont
 StartupPreferencesLoader -> #AllowStartupScript
 StringMorph -> #EditableStringMorph
 SystemProgressMorph -> #horizontalPosition
 SystemProgressMorph -> #verticalPosition
 SystemWindow -> #CloseBoxImage
 SystemWindow -> #CollapseBoxImage
 SystemWindow -> #FullscreenMargin
 TaskListMorph -> #KeepOpen
 TaskbarMorph -> #ShowTaskbar
 TaskbarMorph -> #ShowWindowPreview
 TextDiffBuilder -> #IgnoreLineEndings
 TextDiffBuilder -> #InsertTextAttributes
 TextDiffBuilder -> #NormalTextAttributes
 TextDiffBuilder -> #RemoveTextAttributes
 TextEditor -> #CaseSensitiveFinds

TextEditor -> #UseFindReplaceSelection
 TextEditor -> #UseSecondarySelection
 TextEditor -> #UseSelectionBar
 TextEditor -> #cmdActions
 TextEditor -> #shiftCmdActions
 TextEntryDialogWindow -> #MinimumWidth
 UITheme -> #defaultSettings
 UserInterruptHandler -> #CmdDotEnabled
 Week -> #StartDay
 WorldState -> #CanSurrenderToOS
 WorldState -> #DebugShowDamage
 WorldState -> #DesktopMenuPragmaKeyword
 WorldState -> #DesktopMenuTitle
 WorldState -> #EasySelectingWorld
 WorldState -> #MinCycleLapse
 WorldState -> #ServerMode
 WorldState -> #ShowUpdateOptionInWorldMenu
 ZnConstants -> #DefaultMaximumEntitySize
 ZnServer -> #AlwaysRestart

A.9 Constants

AJConstants -> #CcA
 AJConstants -> #CcABOVE
 AJConstants -> #CcABOVEEQUAL
 AJConstants -> #CcAE
 AJConstants -> #CcB
 AJConstants -> #CcBE
 AJConstants -> #CcBELOW
 AJConstants -> #CcBELOWEQUAL
 AJConstants -> #CcC
 AJConstants -> #CcE
 AJConstants -> #CcEQUAL
 AJConstants -> #CcFPNOTUNORDERED
 AJConstants -> #CcFPUNORDERED
 AJConstants -> #CcG
 AJConstants -> #CcGE
 AJConstants -> #CcGREATER
 AJConstants -> #CcGREATEREQUAL
 AJConstants -> #CcL
 AJConstants -> #CcLE
 AJConstants -> #CcLESS
 AJConstants -> #CcLESSEQUAL
 AJConstants -> #CcNA
 AJConstants -> #CcNAE
 AJConstants -> #CcNB
 AJConstants -> #CcNBE
 AJConstants -> #CcNC
 AJConstants -> #CcNE
 AJConstants -> #CcNEGATIVE
 AJConstants -> #CcNG
 AJConstants -> #CcNGE
 AJConstants -> #CcNL
 AJConstants -> #CcNLE
 AJConstants -> #CcNO
 AJConstants -> #CcNOCONDITION
 AJConstants -> #CcNOOVERFLOW
 AJConstants -> #CcNOTEQUAL
 AJConstants -> #CcNOTSIGN
 AJConstants -> #CcNOTZERO

AJConstants -> #CcNP
 AJConstants -> #CcNS
 AJConstants -> #CcNZ
 AJConstants -> #CcO
 AJConstants -> #CcOVERFLOW
 AJConstants -> #CcP
 AJConstants -> #CcPARITYEVEN
 AJConstants -> #CcPARITYODD
 AJConstants -> #CcPE
 AJConstants -> #CcPO
 AJConstants -> #CcPOSITIVE
 AJConstants -> #CcS
 AJConstants -> #CcSIGN
 AJConstants -> #CcZ
 AJConstants -> #CcZERO
 AJConstants -> #InstCMOVA
 AJConstants -> #InstJA
 AJConstants -> #O64Only
 AJConstants -> #OFM1
 AJConstants -> #OFM10
 AJConstants -> #OFM2
 AJConstants -> #OFM24
 AJConstants -> #OFM248
 AJConstants -> #OFM4
 AJConstants -> #OFM48
 AJConstants -> #OFM4810
 AJConstants -> #OFM8
 AJConstants -> #OG16
 AJConstants -> #OG163264
 AJConstants -> #OG32
 AJConstants -> #OG3264
 AJConstants -> #OG64
 AJConstants -> #OG8
 AJConstants -> #OG8163264
 AJConstants -> #OIMM
 AJConstants -> #OMEM
 AJConstants -> #OMM
 AJConstants -> #OMMMEM
 AJConstants -> #OMMXMM
 AJConstants -> #OMMXMMMEM
 AJConstants -> #ONOREX
 AJConstants -> #OXMM
 AJConstants -> #OXMMMEM
 AJConstants -> #OpImm
 AJConstants -> #OpLabel
 AJConstants -> #OpMem
 AJConstants -> #OpNONE
 AJConstants -> #OpREG
 AJConstants -> #PrefetchNTA
 AJConstants -> #PrefetchT0
 AJConstants -> #PrefetchT1
 AJConstants -> #PrefetchT2
 AJConstants -> #RIDEAX
 AJConstants -> #RIDEBP
 AJConstants -> #RIDE BX
 AJConstants -> #RIDE CX
 AJConstants -> #RIDE DI
 AJConstants -> #RIDE DX
 AJConstants -> #RIDE SI

AJConstants -> #RIDESP	AJx86Registers -> #MM6
AJConstants -> #RegCodeMask	AJx86Registers -> #MM7
AJConstants -> #RegGPB	AJx86Registers -> #R10
AJConstants -> #RegGPD	AJx86Registers -> #R10B
AJConstants -> #RegGPQ	AJx86Registers -> #R10D
AJConstants -> #RegGPW	AJx86Registers -> #R10W
AJConstants -> #RegHighByteMask	AJx86Registers -> #R11
AJConstants -> #RegMM	AJx86Registers -> #R11B
AJConstants -> #RegProhibitsRexMask	AJx86Registers -> #R11D
AJConstants -> #RegRequiresRexMask	AJx86Registers -> #R11W
AJConstants -> #RegTypeMask	AJx86Registers -> #R12
AJConstants -> #RegX87	AJx86Registers -> #R12B
AJConstants -> #RegXMM	AJx86Registers -> #R12D
AJConstants -> #SegmentCS	AJx86Registers -> #R12W
AJConstants -> #SegmentDS	AJx86Registers -> #R13
AJConstants -> #SegmentES	AJx86Registers -> #R13B
AJConstants -> #SegmentFS	AJx86Registers -> #R13D
AJConstants -> #SegmentGS	AJx86Registers -> #R13W
AJConstants -> #SegmentNONE	AJx86Registers -> #R14
AJConstants -> #SegmentSS	AJx86Registers -> #R14B
AJConstants -> #SizeByte	AJx86Registers -> #R14D
AJConstants -> #SizeDQWord	AJx86Registers -> #R14W
AJConstants -> #SizeDWord	AJx86Registers -> #R15
AJConstants -> #SizeQWord	AJx86Registers -> #R15B
AJConstants -> #SizeTWord	AJx86Registers -> #R15D
AJConstants -> #SizeWord	AJx86Registers -> #R15W
AJx86InstructionDescription -> #instructions	AJx86Registers -> #R8
AJx86Registers -> #AH	AJx86Registers -> #R8B
AJx86Registers -> #AL	AJx86Registers -> #R8D
AJx86Registers -> #AX	AJx86Registers -> #R8W
AJx86Registers -> #BH	AJx86Registers -> #R9
AJx86Registers -> #BL	AJx86Registers -> #R9B
AJx86Registers -> #BP	AJx86Registers -> #R9D
AJx86Registers -> #BPL	AJx86Registers -> #R9W
AJx86Registers -> #BX	AJx86Registers -> #RAX
AJx86Registers -> #CH	AJx86Registers -> #RBP
AJx86Registers -> #CL	AJx86Registers -> #RBX
AJx86Registers -> #CX	AJx86Registers -> #RCX
AJx86Registers -> #Codes	AJx86Registers -> #RDI
AJx86Registers -> #DH	AJx86Registers -> #RDX
AJx86Registers -> #DI	AJx86Registers -> #RIP
AJx86Registers -> #DIL	AJx86Registers -> #RSI
AJx86Registers -> #DL	AJx86Registers -> #RSP
AJx86Registers -> #DX	AJx86Registers -> #SI
AJx86Registers -> #EAX	AJx86Registers -> #SIL
AJx86Registers -> #EBP	AJx86Registers -> #SP
AJx86Registers -> #EBX	AJx86Registers -> #SPL
AJx86Registers -> #ECX	AJx86Registers -> #ST0
AJx86Registers -> #EDI	AJx86Registers -> #ST1
AJx86Registers -> #EDX	AJx86Registers -> #ST2
AJx86Registers -> #EIP	AJx86Registers -> #ST3
AJx86Registers -> #ESI	AJx86Registers -> #ST4
AJx86Registers -> #ESP	AJx86Registers -> #ST5
AJx86Registers -> #IP	AJx86Registers -> #ST6
AJx86Registers -> #MM0	AJx86Registers -> #ST7
AJx86Registers -> #MM1	AJx86Registers -> #XMM0
AJx86Registers -> #MM2	AJx86Registers -> #XMM1
AJx86Registers -> #MM3	AJx86Registers -> #XMM10
AJx86Registers -> #MM4	AJx86Registers -> #XMM11
AJx86Registers -> #MM5	

AJx86Registers -> #XMM12
 AJx86Registers -> #XMM13
 AJx86Registers -> #XMM14
 AJx86Registers -> #XMM15
 AJx86Registers -> #XMM2
 AJx86Registers -> #XMM3
 AJx86Registers -> #XMM4
 AJx86Registers -> #XMM5
 AJx86Registers -> #XMM6
 AJx86Registers -> #XMM7
 AJx86Registers -> #XMM8
 AJx86Registers -> #XMM9
 AsyncFile -> #Busy
 AsyncFile -> #ErrorCode
 AthensBezierConverter -> #CollinearityEps
 AthensBezierConverter -> #CurveAngleTolerance
 AthensBezierConverter -> #DistanceEps
 AthensCairoDefs -> #CAIRO_ANTIALIAS_BEST
 AthensCairoDefs -> #CAIRO_ANTIALIAS_DEFAULT
 AthensCairoDefs -> #CAIRO_ANTIALIAS_FAST
 AthensCairoDefs -> #CAIRO_ANTIALIAS_GOOD
 AthensCairoDefs -> #CAIRO_ANTIALIAS_GRAY
 AthensCairoDefs -> #CAIRO_ANTIALIAS_NONE
 AthensCairoDefs -> #CAIRO_ANTIALIAS_SUBPIXEL
 AthensCairoDefs -> #CAIRO_EXTEND_NONE
 AthensCairoDefs -> #CAIRO_EXTEND_PAD
 AthensCairoDefs -> #CAIRO_EXTEND_REFLECT
 AthensCairoDefs -> #CAIRO_EXTEND_REPEAT
 AthensCairoDefs -> #CAIRO_FONT_SLANT_ITALIC
 AthensCairoDefs -> #CAIRO_FONT_SLANT_NORMAL
 AthensCairoDefs -> #CAIRO_FONT_SLANT_OBLIQUE
 AthensCairoDefs -> #CAIRO_FONT_TYPE_FT
 AthensCairoDefs -> #CAIRO_FONT_TYPE_QUARTZ
 AthensCairoDefs -> #CAIRO_FONT_TYPE_TOY
 AthensCairoDefs -> #CAIRO_FONT_TYPE_USER
 AthensCairoDefs -> #CAIRO_FONT_TYPE_WIN32
 AthensCairoDefs -> #CAIRO_FONT_WEIGHT_BOLD
 AthensCairoDefs -> #CAIRO_FONT_WEIGHT_NORMAL
 AthensCairoDefs -> #CAIRO_FORMAT_A1
 AthensCairoDefs -> #CAIRO_FORMAT_A8
 AthensCairoDefs -> #CAIRO_FORMAT_ARGB32
 AthensCairoDefs -> #CAIRO_FORMAT_INVALID
 AthensCairoDefs -> #CAIRO_FORMAT_RGB16_565
 AthensCairoDefs -> #CAIRO_FORMAT_RGB24
 AthensCairoDefs -> #CAIRO_HINT_METRICS_DEFAULT
 AthensCairoDefs -> #CAIRO_HINT_METRICS_OFF
 AthensCairoDefs -> #CAIRO_HINT_METRICS_ON
 AthensCairoDefs -> #CAIRO_HINT_STYLE_DEFAULT
 AthensCairoDefs -> #CAIRO_HINT_STYLE_FULL
 AthensCairoDefs -> #CAIRO_HINT_STYLE_MEDIUM
 AthensCairoDefs -> #CAIRO_HINT_STYLE_NONE
 AthensCairoDefs -> #CAIRO_HINT_STYLE_SLIGHT
 AthensCairoDefs -> #CAIRO_LINE_CAP_BUTT
 AthensCairoDefs -> #CAIRO_LINE_CAP_ROUND
 AthensCairoDefs -> #CAIRO_LINE_CAP_SQUARE
 AthensCairoDefs -> #CAIRO_LINE_JOIN_BEVEL
 AthensCairoDefs -> #CAIRO_LINE_JOIN_MITER
 AthensCairoDefs -> #CAIRO_LINE_JOIN_ROUND
 AthensCairoDefs -> #CAIRO_OPERATOR_ADD
 AthensCairoDefs -> #CAIRO_OPERATOR_ATOP
 AthensCairoDefs -> #CAIRO_OPERATOR_CLEAR
 AthensCairoDefs -> #CAIRO_OPERATOR_COLOR_BURN
 AthensCairoDefs -> #CAIRO_OPERATOR_COLOR_DODGE
 AthensCairoDefs -> #CAIRO_OPERATOR_DARKEN
 AthensCairoDefs -> #CAIRO_OPERATOR_DEST
 AthensCairoDefs -> #CAIRO_OPERATOR_DEST_ATOP
 AthensCairoDefs -> #CAIRO_OPERATOR_DEST_IN
 AthensCairoDefs -> #CAIRO_OPERATOR_DEST_OUT
 AthensCairoDefs -> #CAIRO_OPERATOR_DEST_OVER
 AthensCairoDefs -> #CAIRO_OPERATOR_DIFFERENCE
 AthensCairoDefs -> #CAIRO_OPERATOR_EXCLUSION
 AthensCairoDefs -> #CAIRO_OPERATOR_HARD_LIGHT
 AthensCairoDefs -> #CAIRO_OPERATOR_HSL_COLOR
 AthensCairoDefs -> #CAIRO_OPERATOR_HSL_HUE
 AthensCairoDefs -> #CAIRO_OPERATOR_HSL_LUMINOSITY
 AthensCairoDefs -> #CAIRO_OPERATOR_HSL_SATURATION
 AthensCairoDefs -> #CAIRO_OPERATOR_IN
 AthensCairoDefs -> #CAIRO_OPERATOR_LIGHTEN
 AthensCairoDefs -> #CAIRO_OPERATOR_MULTIPLY
 AthensCairoDefs -> #CAIRO_OPERATOR_OUT
 AthensCairoDefs -> #CAIRO_OPERATOR_OVER
 AthensCairoDefs -> #CAIRO_OPERATOR_OVERLAY
 AthensCairoDefs -> #CAIRO_OPERATOR_SATURATE
 AthensCairoDefs -> #CAIRO_OPERATOR_SCREEN
 AthensCairoDefs -> #CAIRO_OPERATOR_SOFT_LIGHT
 AthensCairoDefs -> #CAIRO_OPERATOR_SOURCE
 AthensCairoDefs -> #CAIRO_OPERATOR_XOR
 AthensCairoDefs -> #CAIRO_STATUS_CLIP_NOT_REPRESENTABLE
 AthensCairoDefs -> #CAIRO_STATUS_DEVICE_ERROR
 AthensCairoDefs -> #CAIRO_STATUS_DEVICE_TYPE_MISMATCH
 AthensCairoDefs -> #CAIRO_STATUS_FILE_NOT_FOUND
 AthensCairoDefs -> #CAIRO_STATUS_FONT_TYPE_MISMATCH
 AthensCairoDefs -> #CAIRO_STATUS_INVALID_CLUSTERS
 AthensCairoDefs -> #CAIRO_STATUS_INVALID_CONTENT
 AthensCairoDefs -> #CAIRO_STATUS_INVALID_DASH
 AthensCairoDefs -> #CAIRO_STATUS_INVALID_DSC_COMMENT
 AthensCairoDefs -> #CAIRO_STATUS_INVALID_FORMAT
 AthensCairoDefs -> #CAIRO_STATUS_INVALID_INDEX
 AthensCairoDefs -> #CAIRO_STATUS_INVALID_MATRIX
 AthensCairoDefs -> #CAIRO_STATUS_INVALID_PATH_DATA
 AthensCairoDefs -> #CAIRO_STATUS_INVALID_POP_GROUP
 AthensCairoDefs -> #CAIRO_STATUS_INVALID_RESTORE
 AthensCairoDefs -> #CAIRO_STATUS_INVALID_SIZE
 AthensCairoDefs -> #CAIRO_STATUS_INVALID_SLANT
 AthensCairoDefs -> #CAIRO_STATUS_INVALID_STATUS
 AthensCairoDefs -> #CAIRO_STATUS_INVALID_STRIDE
 AthensCairoDefs -> #CAIRO_STATUS_INVALID_STRING
 AthensCairoDefs -> #CAIRO_STATUS_INVALID_VISUAL
 AthensCairoDefs -> #CAIRO_STATUS_INVALID_WEIGHT
 AthensCairoDefs -> #CAIRO_STATUS_LAST_STATUS
 AthensCairoDefs -> #CAIRO_STATUS_NEGATIVE_COUNT
 AthensCairoDefs -> #CAIRO_STATUS_NO_CURRENT_POINT
 AthensCairoDefs -> #CAIRO_STATUS_NO_MEMORY
 AthensCairoDefs -> #CAIRO_STATUS_NULL_POINTER
 AthensCairoDefs -> #CAIRO_STATUS_PATTERN_TYPE_MISMATCH
 AthensCairoDefs -> #CAIRO_STATUS_READ_ERROR
 AthensCairoDefs -> #CAIRO_STATUS_SUCCESS
 AthensCairoDefs -> #CAIRO_STATUS_SURFACE_FINISHED
 AthensCairoDefs -> #CAIRO_STATUS_SURFACE_TYPE_MISMATCH

AthensCairoDefs -> #CAIRO_STATUS_TEMP_FILE_ERROR
 AthensCairoDefs -> #CAIRO_STATUS_USER_FONT_ERROR
 AthensCairoDefs -> #CAIRO_STATUS_USER_FONT_IMMUTABLE
 AthensCairoDefs -> #CAIRO_STATUS_USER_FONT_NOT_IMPLEMENTED
 AthensCairoDefs -> #CAIRO_STATUS_WRITE_ERROR
 AthensCairoDefs -> #CAIRO_SUBPIXEL_ORDER_BGR
 AthensCairoDefs -> #CAIRO_SUBPIXEL_ORDER_DEFAULT
 AthensCairoDefs -> #CAIRO_SUBPIXEL_ORDER_RGB
 AthensCairoDefs -> #CAIRO_SUBPIXEL_ORDER_VBGR
 AthensCairoDefs -> #CAIRO_SUBPIXEL_ORDER_VRGB
 AthensCairoDefs -> #cairo_font_slant_t
 AthensCairoDefs -> #cairo_font_type_t
 AthensCairoDefs -> #cairo_font_weight_t
 AthensCairoDefs -> #cairo_line_cap_t
 AthensCairoDefs -> #cairo_line_join_t
 AthensCairoDefs -> #cairo_operator_t
 AthensCairoDefs -> #cairo_pattern_t
 AthensCairoDefs -> #cairo_status_t
 AthensCairoDefs -> #cairo_surface_t
 AthensCairoDefs -> #cairo_t
 AthensCairoDefs -> #cairo_text_extents_t
 AthensCurveFlattener -> #CollinearityEps
 AthensCurveFlattener -> #CurveAngleToleranceEpsilon
 AthensCurveFlattener -> #CurveCollinearityEpsilon
 AthensCurveFlattener -> #CurveDistanceEpsilon
 AthensCurveFlattener -> #SubdivisionLimit
 AthensPathBuilder -> #ZeroPoint
 BalloonEngineConstants -> #BEAaLevelIndex
 BalloonEngineConstants -> #BEBalloonEngineSize
 BalloonEngineConstants -> #BEBitBlitIndex
 BalloonEngineConstants -> #BEClipRectIndex
 BalloonEngineConstants -> #BECColorTransformIndex
 BalloonEngineConstants -> #BEDeferredIndex
 BalloonEngineConstants -> #BEDestOffsetIndex
 BalloonEngineConstants -> #BEEdgeTransformIndex
 BalloonEngineConstants -> #BEEexternalsIndex
 BalloonEngineConstants -> #BEFormsIndex
 BalloonEngineConstants -> #BEPstFlushNeededIndex
 BalloonEngineConstants -> #BESpanIndex
 BalloonEngineConstants -> #BEWorkBufferIndex
 BalloonEngineConstants -> #ETBalloonEdgeDataSize
 BalloonEngineConstants -> #ETIndexIndex
 BalloonEngineConstants -> #ETLinesIndex
 BalloonEngineConstants -> #ETSourceIndex
 BalloonEngineConstants -> #ETXValueIndex
 BalloonEngineConstants -> #ETYValueIndex
 BalloonEngineConstants -> #ETZValueIndex
 BalloonEngineConstants -> #FTBalloonFillDataSize
 BalloonEngineConstants -> #FTDestFormIndex
 BalloonEngineConstants -> #FTIndexIndex
 BalloonEngineConstants -> #FTMaxXIndex
 BalloonEngineConstants -> #FTMinXIndex
 BalloonEngineConstants -> #FTSourceIndex
 BalloonEngineConstants -> #FTYValueIndex
 BalloonEngineConstants -> #GBaseSize
 BalloonEngineConstants -> #GBaseDepth
 BalloonEngineConstants -> #GBaseHeight
 BalloonEngineConstants -> #GBaseRaster
 BalloonEngineConstants -> #GBaseSize
 BalloonEngineConstants -> #GBaseWidth
 BalloonEngineConstants -> #GColorMapOffset
 BalloonEngineConstants -> #GColorMapSize
 BalloonEngineConstants -> #GEndX
 BalloonEngineConstants -> #GEndY
 BalloonEngineConstants -> #GFinalX
 BalloonEngineConstants -> #GMBBaseSize
 BalloonEngineConstants -> #GBTileFlag
 BalloonEngineConstants -> #GBUpdateDDX
 BalloonEngineConstants -> #GBUpdateDDY
 BalloonEngineConstants -> #GBUpdateDX
 BalloonEngineConstants -> #GBUpdateDY
 BalloonEngineConstants -> #GBUpdateData
 BalloonEngineConstants -> #GBUpdateX
 BalloonEngineConstants -> #GBUpdateY
 BalloonEngineConstants -> #GBViaX
 BalloonEngineConstants -> #GBViaY
 BalloonEngineConstants -> #GBWideEntry
 BalloonEngineConstants -> #GBWideExit
 BalloonEngineConstants -> #GBWideExtent
 BalloonEngineConstants -> #GBWideFill
 BalloonEngineConstants -> #GBWideSize
 BalloonEngineConstants -> #GBWideUpdateData
 BalloonEngineConstants -> #GBWideWidth
 BalloonEngineConstants -> #GBaseEdgeSize
 BalloonEngineConstants -> #GBaseFillSize
 BalloonEngineConstants -> #GEdgeClipFlag
 BalloonEngineConstants -> #GEdgeFillsInvalid
 BalloonEngineConstants -> #GFillIndexLeft
 BalloonEngineConstants -> #GFillIndexRight
 BalloonEngineConstants -> #GNumLines
 BalloonEngineConstants -> #GObjectIndex
 BalloonEngineConstants -> #GObjectLength
 BalloonEngineConstants -> #GObjectType
 BalloonEngineConstants -> #GObjectUnused
 BalloonEngineConstants -> #GPrimitiveBezier
 BalloonEngineConstants -> #GPrimitiveClippedBitmapFill
 BalloonEngineConstants -> #GPrimitiveEdge
 BalloonEngineConstants -> #GPrimitiveEdgeMask
 BalloonEngineConstants -> #GPrimitiveFill
 BalloonEngineConstants -> #GPrimitiveFillMask
 BalloonEngineConstants -> #GPrimitiveLine
 BalloonEngineConstants -> #GPrimitiveLinearGradientFill
 BalloonEngineConstants -> #GPrimitiveRadialGradientFill
 BalloonEngineConstants -> #GPrimitiveRepeatedBitmapFill
 BalloonEngineConstants -> #GPrimitiveTypeMask
 BalloonEngineConstants -> #GPrimitiveUnknown
 BalloonEngineConstants -> #GPrimitiveWide
 BalloonEngineConstants -> #GPrimitiveWideBezier
 BalloonEngineConstants -> #GPrimitiveWideEdge
 BalloonEngineConstants -> #GPrimitiveWideLine
 BalloonEngineConstants -> #GPrimitiveWideMask
 BalloonEngineConstants -> #GStateAddingFromGET
 BalloonEngineConstants -> #GStateBlitBuffer
 BalloonEngineConstants -> #GStateCompleted
 BalloonEngineConstants -> #GStateScanningAET
 BalloonEngineConstants -> #GStateUnlocked
 BalloonEngineConstants -> #GStateUpdateEdges
 BalloonEngineConstants -> #GStateWaitingChange
 BalloonEngineConstants -> #GStateWaitingForEdge

BalloonEngineConstants -> #GStateWaitingForFill
 BalloonEngineConstants -> #GEXValue
 BalloonEngineConstants -> #GEYValue
 BalloonEngineConstants -> #GEZValue
 BalloonEngineConstants -> #GErrorAETEntry
 BalloonEngineConstants -> #GErrorBadState
 BalloonEngineConstants -> #GErrorFillEntry
 BalloonEngineConstants -> #GErrorGETEntry
 BalloonEngineConstants -> #GErrorNeedFlush
 BalloonEngineConstants -> #GErrorNoMoreSpace
 BalloonEngineConstants -> #GFDirectionX
 BalloonEngineConstants -> #GFDirectionY
 BalloonEngineConstants -> #GFNormalX
 BalloonEngineConstants -> #GFNormalY
 BalloonEngineConstants -> #GFOriginX
 BalloonEngineConstants -> #GFOriginY
 BalloonEngineConstants -> #GFRampLength
 BalloonEngineConstants -> #GFRampOffset
 BalloonEngineConstants -> #GGBaseSize
 BalloonEngineConstants -> #GLBaseSize
 BalloonEngineConstants -> #GLEndX
 BalloonEngineConstants -> #GLEndY
 BalloonEngineConstants -> #GLError
 BalloonEngineConstants -> #GLErrorAdjDown
 BalloonEngineConstants -> #GLErrorAdjUp
 BalloonEngineConstants -> #GLWideEntry
 BalloonEngineConstants -> #GLWideExit
 BalloonEngineConstants -> #GLWideExtent
 BalloonEngineConstants -> #GLWideFill
 BalloonEngineConstants -> #GLWideSize
 BalloonEngineConstants -> #GLWideWidth
 BalloonEngineConstants -> #GLXDirection
 BalloonEngineConstants -> #GLXIncrement
 BalloonEngineConstants -> #GLYDirection
 BalloonEngineConstants -> #GWAAColorMask
 BalloonEngineConstants -> #GWAAColorShift
 BalloonEngineConstants -> #GWAALHalfPixel
 BalloonEngineConstants -> #GWAALLevel
 BalloonEngineConstants -> #GWAAScanMask
 BalloonEngineConstants -> #GWAAShift
 BalloonEngineConstants -> #GWAETStart
 BalloonEngineConstants -> #GWAETUsed
 BalloonEngineConstants -> #GWBezierHeightSubdivisions
 BalloonEngineConstants -> #GWBezierLineConversions
 BalloonEngineConstants -> #GWBezierMonotonSubdivisions
 BalloonEngineConstants -> #GWBezierOverflowSubdivisions
 BalloonEngineConstants -> #GWBufferTop
 BalloonEngineConstants -> #GWClearSpanBuffer
 BalloonEngineConstants -> #GWClipMaxX
 BalloonEngineConstants -> #GWClipMaxY
 BalloonEngineConstants -> #GWClipMinX
 BalloonEngineConstants -> #GWClipMinY
 BalloonEngineConstants -> #GWColorTransform
 BalloonEngineConstants -> #GWCountAddAETEntry
 BalloonEngineConstants -> #GWCountChangeAETEntry
 BalloonEngineConstants -> #GWCountDisplaySpan
 BalloonEngineConstants -> #GWCountFinishTest
 BalloonEngineConstants -> #GWCountInitializing
 BalloonEngineConstants -> #GWCountMergeFill
 BalloonEngineConstants -> #GWCountNextAETEntry
 BalloonEngineConstants -> #GWCountNextFillEntry
 BalloonEngineConstants -> #GWCountNextGETEntry
 BalloonEngineConstants -> #GWCurrentY
 BalloonEngineConstants -> #GWCurrentZ
 BalloonEngineConstants -> #GWDestOffsetX
 BalloonEngineConstants -> #GWDestOffsetY
 BalloonEngineConstants -> #GWEdgeTransform
 BalloonEngineConstants -> #GWFillMaxX
 BalloonEngineConstants -> #GWFillMaxY
 BalloonEngineConstants -> #GWFillMinX
 BalloonEngineConstants -> #GWFillMinY
 BalloonEngineConstants -> #GWFillOffsetX
 BalloonEngineConstants -> #GWFillOffsetY
 BalloonEngineConstants -> #GWGETStart
 BalloonEngineConstants -> #GWGETUsed
 BalloonEngineConstants -> #GWHasClipShapes
 BalloonEngineConstants -> #GWHasColorTransform
 BalloonEngineConstants -> #GWHasEdgeTransform
 BalloonEngineConstants -> #GWHeaderSize
 BalloonEngineConstants -> #GWLastExportedEdge
 BalloonEngineConstants -> #GWLastExportedFill
 BalloonEngineConstants -> #GWLastExportedLeftX
 BalloonEngineConstants -> #GWLastExportedRightX
 BalloonEngineConstants -> #GWMagicIndex
 BalloonEngineConstants -> #GWMagicNumber
 BalloonEngineConstants -> #GWMinimalSize
 BalloonEngineConstants -> #GWNeedsFlush
 BalloonEngineConstants -> #GWObjStart
 BalloonEngineConstants -> #GWObjUsed
 BalloonEngineConstants -> #GWPoint1
 BalloonEngineConstants -> #GWPoint2
 BalloonEngineConstants -> #GWPoint3
 BalloonEngineConstants -> #GWPoint4
 BalloonEngineConstants -> #GWPointListFirst
 BalloonEngineConstants -> #GWSize
 BalloonEngineConstants -> #GWSpanEnd
 BalloonEngineConstants -> #GWSpanEndAA
 BalloonEngineConstants -> #GWSpanSize
 BalloonEngineConstants -> #GWSpanStart
 BalloonEngineConstants -> #GWState
 BalloonEngineConstants -> #GWStopReason
 BalloonEngineConstants -> #GWTimeAddAETEntry
 BalloonEngineConstants -> #GWTimeChangeAETEntry
 BalloonEngineConstants -> #GWTimeDisplaySpan
 BalloonEngineConstants -> #GWTimeFinishTest
 BalloonEngineConstants -> #GWTimeInitializing
 BalloonEngineConstants -> #GWTimeMergeFill
 BalloonEngineConstants -> #GWTimeNextAETEntry
 BalloonEngineConstants -> #GWTimeNextFillEntry
 BalloonEngineConstants -> #GWTimeNextGETEntry
 Base64MimeConverter -> #FromCharTable
 Base64MimeConverter -> #ToCharTable
 ByteString -> #NonAsciiMap
 ByteTextConverter -> #byteToUnicode
 ByteTextConverter -> #unicodeToByte
 Categorizer -> #Default
 Categorizer -> #NullCategory
 Character -> #CharacterTable
 Character -> #DigitValues

CharacterScanner -> #ColumnBreakStopConditions
 CharacterScanner -> #CompositionStopConditions
 CharacterScanner -> #DefaultStopConditions
 CharacterScanner -> #MeasuringStopConditions
 CharacterScanner -> #PaddedSpaceCondition
 CharacterSet -> #CrLf
 ChronologyConstants -> #DayNames
 ChronologyConstants -> #DaysInMonth
 ChronologyConstants -> #MicrosecondsInDay
 ChronologyConstants -> #MonthNames
 ChronologyConstants -> #NanosInMillisecond
 ChronologyConstants -> #NanosInSecond
 ChronologyConstants -> #SecondsInDay
 ChronologyConstants -> #SecondsInHour
 ChronologyConstants -> #SecondsInMinute
 ChronologyConstants -> #SqueakEpoch
 Color -> #BlueShift
 Color -> #ColorRegistry
 Color -> #ComponentMask
 Color -> #ComponentMax
 Color -> #GrayToIndexMap
 Color -> #GreenShift
 Color -> #HalfComponentMask
 Color -> #IndexedColors
 Color -> #RedShift
 ColorPresenterMorph -> #HatchForm
 CombinedChar -> #Compositions
 CombinedChar -> #Decompositions
 CombinedChar -> #Diacriticals
 CompiledMethod -> #LargeFrame
 CompiledMethod -> #SmallFrame
 ContextPart -> #PrimitiveFailToken
 ContextPart -> #SpecialPrimitiveSimulators
 ContextPart -> #TryNamedPrimitiveTemplateMethod
 CornerRounder -> #CR0
 CornerRounder -> #CR1
 CornerRounder -> #CR2
 Cursor -> #BlankCursor
 Cursor -> #BottomLeftCursor
 Cursor -> #BottomRightCursor
 Cursor -> #CornerCursor
 Cursor -> #CrossHairCursor
 Cursor -> #CurrentCursor
 Cursor -> #DownCursor
 Cursor -> #MarkerCursor
 Cursor -> #MenuCursor
 Cursor -> #MoveCursor
 Cursor -> #NormalCursor
 Cursor -> #OriginCursor
 Cursor -> #OverEditableText
 Cursor -> #ReadCursor
 Cursor -> #ResizeLeftCursor
 Cursor -> #ResizeTopCursor
 Cursor -> #ResizeTopLeftCursor
 Cursor -> #ResizeTopRightCursor
 Cursor -> #RightArrowCursor
 Cursor -> #SquareCursor
 Cursor -> #TargetCursor
 Cursor -> #TopLeftCursor
 Cursor -> #TopRightCursor
 Cursor -> #UpCursor
 Cursor -> #WaitCursor
 Cursor -> #WebLinkCursor
 Cursor -> #WriteCursor
 Cursor -> #XeqCursor
 Decompiler -> #ArgumentFlag
 Decompiler -> #CascadeFlag
 Decompiler -> #CaseFlag
 Decompiler -> #IfNilFlag
 DigitalSignatureAlgorithm -> #HighBitOfByte
 DigitalSignatureAlgorithm -> #SmallPrimes
 DisplayMedium -> #HighLightBitmaps
 EventSensorConstants -> #BlueButtonBit
 EventSensorConstants -> #CommandKeyBit
 EventSensorConstants -> #CtrlKeyBit
 EventSensorConstants -> #EventKeyChar
 EventSensorConstants -> #EventKeyDown
 EventSensorConstants -> #EventKeyUp
 EventSensorConstants -> #EventTypeDragDropFiles
 EventSensorConstants -> #EventTypeKeyboard
 EventSensorConstants -> #EventTypeMenu
 EventSensorConstants -> #EventTypeMouse
 EventSensorConstants -> #EventTypeNone
 EventSensorConstants -> #EventTypeWindow
 EventSensorConstants -> #OptionKeyBit
 EventSensorConstants -> #RedButtonBit
 EventSensorConstants -> #ShiftKeyBit
 EventSensorConstants -> #WindowEventActivated
 EventSensorConstants -> #WindowEventClose
 EventSensorConstants -> #WindowEventIconise
 EventSensorConstants -> #WindowEventMetricChange
 EventSensorConstants -> #WindowEventPaint
 EventSensorConstants -> #YellowButtonBit
 FLLargeIdentityHashedCollection -> #PermutationMap
 FT2Constants -> #LoadCropBitmap
 FT2Constants -> #LoadDefault
 FT2Constants -> #LoadForceAutohint
 FT2Constants -> #LoadIgnoreGlobalAdvanceWidth
 FT2Constants -> #LoadIgnoreTransform
 FT2Constants -> #LoadLinearDesign
 FT2Constants -> #LoadMonochrome
 FT2Constants -> #LoadNoAutohint
 FT2Constants -> #LoadNoBitmap
 FT2Constants -> #LoadNoHinting
 FT2Constants -> #LoadNoRecurse
 FT2Constants -> #LoadNoScale
 FT2Constants -> #LoadPedantic
 FT2Constants -> #LoadRender
 FT2Constants -> #LoadSbitsOnly
 FT2Constants -> #LoadTargetLCD
 FT2Constants -> #LoadTargetLCDV
 FT2Constants -> #LoadTargetLight
 FT2Constants -> #LoadTargetMono
 FT2Constants -> #LoadTargetNormal
 FT2Constants -> #LoadVerticalLayout
 FT2Constants -> #PixelModeGray
 FT2Constants -> #PixelModeGray2
 FT2Constants -> #PixelModeGray4
 FT2Constants -> #PixelModeLCD

FT2Constants -> #PixelModeLCDV
 FT2Constants -> #PixelModeMono
 FT2Constants -> #PixelModeNone
 FT2Constants -> #RenderModeLCD
 FT2Constants -> #RenderModeLCDV
 FT2Constants -> #RenderModeLight
 FT2Constants -> #RenderModeMono
 FT2Constants -> #RenderModeNormal
 FT2Constants -> #StyleFlagBold
 FT2Constants -> #StyleFlagItalic
 FastInflateStream -> #DistanceMap
 FastInflateStream -> #FixedDistTable
 FastInflateStream -> #FixedLitTable
 FastInflateStream -> #LiteralLengthMap
 Float -> #E
 Float -> #Epsilon
 Float -> #Halfpi
 Float -> #Infinity
 Float -> #Ln10
 Float -> #Ln2
 Float -> #MaxVal
 Float -> #MaxValLn
 Float -> #MinValLogBase2
 Float -> #NaN
 Float -> #NegativeInfinity
 Float -> #NegativeZero
 Float -> #Pi
 Float -> #RadiansPerDegree
 Float -> #Sqrt2
 Float -> #ThreePi
 Float -> #Twopi
 FormCanvas -> #TranslucentPatterns
 FreeTypeCacheConstants -> #FreeTypeCacheGlyph
 FreeTypeCacheConstants -> #FreeTypeCacheGlyphLCD
 FreeTypeCacheConstants -> #FreeTypeCacheGlyphMono
 FreeTypeCacheConstants -> #FreeTypeCacheLinearWidth
 FreeTypeCacheConstants -> #FreeTypeCacheWidth
 FreeTypeNameParser -> #italicNames
 FreeTypeNameParser -> #normalNames
 FreeTypeNameParser -> #obliqueNames
 FreeTypeNameParser -> #stretchNames
 FreeTypeNameParser -> #weightNames
 GIFReadWriter -> #Extension
 GIFReadWriter -> #ImageSeparator
 GIFReadWriter -> #Terminator
 GZipConstants -> #GZipAsciiFlag
 GZipConstants -> #GZipCommentFlag
 GZipConstants -> #GZipContinueFlag
 GZipConstants -> #GZipDeflated
 GZipConstants -> #GZipEncryptFlag
 GZipConstants -> #GZipExtraField
 GZipConstants -> #GZipMagic
 GZipConstants -> #GZipNameFlag
 GZipConstants -> #GZipReservedFlags
 HashTableSizes -> #sizes
 IRBytecodeGenerator -> #BytecodeTable
 IRBytecodeGenerator -> #Bytecodes
 IRBytecodeGenerator -> #SpecialConstants
 IRBytecodeGenerator -> #SpecialSelectors
 ISOLanguageDefinition -> #ISO2Countries

ISOLanguageDefinition -> #ISO2Table
 ISOLanguageDefinition -> #ISO3Countries
 ISOLanguageDefinition -> #ISO3Table
 IconicButton -> #DefaultGraphics
 ImageMorph -> #DefaultForm
 InflateStream -> #BlockProceedBit
 InflateStream -> #BlockTypes
 InflateStream -> #FixedDistCodes
 InflateStream -> #FixedLitCodes
 InflateStream -> #MaxBits
 InflateStream -> #StateNewBlock
 InflateStream -> #StateNoMoreData
 InputEventSensor -> #ButtonDecodeTable
 InstructionStream -> #SpecialConstants
 JPEGHuffmanTable -> #BitBufferSize
 JPEGHuffmanTable -> #Lookahead
 JPEGReadStream -> #MaxBits
 JPEGReadWriter -> #ConstBits
 JPEGReadWriter -> #DCTK1
 JPEGReadWriter -> #DCTK2
 JPEGReadWriter -> #DCTK3
 JPEGReadWriter -> #DCTK4
 JPEGReadWriter -> #DCTSize
 JPEGReadWriter -> #DCTSiz2
 JPEGReadWriter -> #DitherMasks
 JPEGReadWriter -> #FIXn0n298631336
 JPEGReadWriter -> #FIXn0n34414
 JPEGReadWriter -> #FIXn0n390180644
 JPEGReadWriter -> #FIXn0n541196100
 JPEGReadWriter -> #FIXn0n71414
 JPEGReadWriter -> #FIXn0n765366865
 JPEGReadWriter -> #FIXn0n899976223
 JPEGReadWriter -> #FIXn1n175875602
 JPEGReadWriter -> #FIXn1n40200
 JPEGReadWriter -> #FIXn1n501321110
 JPEGReadWriter -> #FIXn1n77200
 JPEGReadWriter -> #FIXn1n847759065
 JPEGReadWriter -> #FIXn1n961570560
 JPEGReadWriter -> #FIXn2n053119869
 JPEGReadWriter -> #FIXn2n562915447
 JPEGReadWriter -> #FIXn3n072711026
 JPEGReadWriter -> #FloatSampleOffset
 JPEGReadWriter -> #HuffmanTableSize
 JPEGReadWriter -> #JFIFMarkerParser
 JPEGReadWriter -> #JPEGNaturalOrder
 JPEGReadWriter -> #MaxSample
 JPEGReadWriter -> #Pass1Bits
 JPEGReadWriter -> #Pass1Div
 JPEGReadWriter -> #Pass2Div
 JPEGReadWriter -> #QTableScaleFactor
 JPEGReadWriter -> #QuantizationTableSize
 JPEGReadWriter -> #SampleOffset
 KMSingleKeyCombination -> #specialKeys
 Key -> #KeyTable
 Key -> #MacosVirtualKeyTable
 Key -> #UnixVirtualKeyTable
 Key -> #WindowsVirtualKeyTable
 KomitClassNode -> #addedClassIcon
 KomitClassNode -> #deletedClassIcon

KomitClassNode -> #modifiedClassIcon
 KomitterUI -> #manageRemotesIcon
 Latin1 -> #rightHalfSequence
 MCDataStream -> #TypeMap
 MCFileTreeStCypressWriter -> #specials
 MD5NonPrimitive -> #ABCDTable
 MD5NonPrimitive -> #IndexTable
 MD5NonPrimitive -> #ShiftTable
 MD5NonPrimitive -> #SinTable
 MailAddressTokenizer -> #CSNonAtom
 MailAddressTokenizer -> #CSNonSeparators
 MailAddressTokenizer -> #CSParens
 MailAddressTokenizer -> #CSSpecials
 MenuItemMorph -> #SubMenuMarker
 MessageNode -> #MacroEmitters
 MessageNode -> #MacroPrinters
 MessageNode -> #MacroSelectors
 MessageNode -> #MacroSizers
 MessageNode -> #MacroTransformers
 MessageNode -> #StdTypers
 MessageNode -> #ThenFlag
 MetacelloVersionValidator -> #reasonCodeDescriptions
 Morph -> #EmptyArray
 MultiByteFileStream -> #Cr
 MultiByteFileStream -> #CrLf
 MultiByteFileStream -> #Lf
 MultiByteFileStream -> #LineEndStrings
 MultiByteFileStream -> #LookAheadCount
 NBFFICallout -> #CustomErrorCodes
 NBFFICallout -> #CustomErrorMessages
 NBFFICallout -> #TypeAliases
 NBInterpreterProxy -> #CogFunctions
 NBInterpreterProxy -> #Functions
 NBMacConstants -> #MAP_ANON
 NBMacConstants -> #MAP_COPY
 NBMacConstants -> #MAP_FAILED
 NBMacConstants -> #MAP_FILE
 NBMacConstants -> #MAP_FIXED
 NBMacConstants -> #MAP_HASSEMAPHORE
 NBMacConstants -> #MAP_NOCACHE
 NBMacConstants -> #MAP_NOEXTEND
 NBMacConstants -> #MAP_NORESERVE
 NBMacConstants -> #MAP_PRIVATE
 NBMacConstants -> #MAP_RENAME
 NBMacConstants -> #MAP_RESERVED0080
 NBMacConstants -> #MAP_SHARED
 NBMacConstants -> #PROT_EXEC
 NBMacConstants -> #PROT_NONE
 NBMacConstants -> #PROT_READ
 NBMacConstants -> #PROT_WRITE
 NBMacConstants -> #RTLD_DEFAULT
 NBMacConstants -> #RTLD_FIRST
 NBMacConstants -> #RTLD_GLOBAL
 NBMacConstants -> #RTLD_LAZY
 NBMacConstants -> #RTLD_LOCAL
 NBMacConstants -> #RTLD_MAIN_ONLY
 NBMacConstants -> #RTLD_NEXT
 NBMacConstants -> #RTLD_NODELETE
 NBMacConstants -> #RTLD_NOLOAD
 NBMacConstants -> #RTLD_NOW
 NBMacConstants -> #RTLD_SELF
 NBUnixConstants -> #MAP_32BIT
 NBUnixConstants -> #MAP_ANON
 NBUnixConstants -> #MAP_ANONYMOUS
 NBUnixConstants -> #MAP_DENYWRITE
 NBUnixConstants -> #MAP_EXECUTABLE
 NBUnixConstants -> #MAP_FAILED
 NBUnixConstants -> #MAP_FILE
 NBUnixConstants -> #MAP_FIXED
 NBUnixConstants -> #MAP_GROWSDOWN
 NBUnixConstants -> #MAP_LOCKED
 NBUnixConstants -> #MAP_NONBLOCK
 NBUnixConstants -> #MAP_NORESERVE
 NBUnixConstants -> #MAP_POPULATE
 NBUnixConstants -> #MAP_PRIVATE
 NBUnixConstants -> #MAP_SHARED
 NBUnixConstants -> #MAP_STACK
 NBUnixConstants -> #MAP_TYPE
 NBUnixConstants -> #PROT_EXEC
 NBUnixConstants -> #PROT_GROWSDOWN
 NBUnixConstants -> #PROT_GROWSUP
 NBUnixConstants -> #PROT_NONE
 NBUnixConstants -> #PROT_READ
 NBUnixConstants -> #PROT_WRITE
 NBUnixConstants -> #RTLD_BINDING_MASK
 NBUnixConstants -> #RTLD_DEEPBIND
 NBUnixConstants -> #RTLD_DEFAULT
 NBUnixConstants -> #RTLD_GLOBAL
 NBUnixConstants -> #RTLD_LAZY
 NBUnixConstants -> #RTLD_LOCAL
 NBUnixConstants -> #RTLD_NEXT
 NBUnixConstants -> #RTLD_NODELETE
 NBUnixConstants -> #RTLD_NOLOAD
 NBUnixConstants -> #RTLD_NOW
 NBWinConstants -> #ABOVE_NORMAL_PRIORITY_CLASS
 NBWinConstants -> #ACCESS_SYSTEM_SECURITY
 NBWinConstants -> #ACE_INHERITED_OBJECT_TYPE_PRESENT
 NBWinConstants -> #ACE_OBJECT_TYPE_PRESENT
 NBWinConstants -> #APPLICATION_ERROR_MASK
 NBWinConstants -> #BELOW_NORMAL_PRIORITY_CLASS
 NBWinConstants -> #CREATE_FORCEDOS
 NBWinConstants -> #CREATE_NEW_CONSOLE
 NBWinConstants -> #CREATE_NEW_PROCESS_GROUP
 NBWinConstants -> #CREATE_SEPARATE_WOW_VDM
 NBWinConstants -> #CREATE_SHARED_WOW_VDM
 NBWinConstants -> #CREATE_SUSPENDED
 NBWinConstants -> #CREATE_UNICODE_ENVIRONMENT
 NBWinConstants -> #CS_BYTEALIGNCLIENT
 NBWinConstants -> #CS_BYTEALIGNWINDOW
 NBWinConstants -> #CS_CLASSDC
 NBWinConstants -> #CS_DBLCLKS
 NBWinConstants -> #CS_DROPSHADOW
 NBWinConstants -> #CS_GLOBALCLASS
 NBWinConstants -> #CS_HREDRAW
 NBWinConstants -> #CS_IME
 NBWinConstants -> #CS_NOCLOSE
 NBWinConstants -> #CS_OWNDC
 NBWinConstants -> #CS_PARENTDC
 NBWinConstants -> #CS_SAVEBITS

NBWinConstants -> #CS_VREDRAW
 NBWinConstants -> #CW_USEDEFAULT
 NBWinConstants -> #DEBUG_ONLY_THIS_PROCESS
 NBWinConstants -> #DEBUG_PROCESS
 NBWinConstants -> #DETACHED_PROCESS
 NBWinConstants -> #DRIVE_CDROM
 NBWinConstants -> #DRIVE_FIXED
 NBWinConstants -> #DRIVE_NO_ROOT_DIR
 NBWinConstants -> #DRIVE_RAMDISK
 NBWinConstants -> #DRIVE_REMOTE
 NBWinConstants -> #DRIVE_REMOVABLE
 NBWinConstants -> #DRIVE_UNKNOWN
 NBWinConstants -> #ERROR_SEVERITY_ERROR
 NBWinConstants -> #ERROR_SEVERITY_INFORMATIONAL
 NBWinConstants -> #ERROR_SEVERITY_SUCCESS
 NBWinConstants -> #ERROR_SEVERITY_WARNING
 NBWinConstants -> #GWL_EXSTYLE
 NBWinConstants -> #GWL_HINSTANCE
 NBWinConstants -> #GWL_HWNDPARENT
 NBWinConstants -> #GWL_ID
 NBWinConstants -> #GWL_STYLE
 NBWinConstants -> #GWL_USERDATA
 NBWinConstants -> #GWL_WNDPROC
 NBWinConstants -> #GW_CHILD
 NBWinConstants -> #GW_ENABLEDPOPUP
 NBWinConstants -> #GW_HWNDFIRST
 NBWinConstants -> #GW_HWNDLAST
 NBWinConstants -> #GW_HWNDNEXT
 NBWinConstants -> #GW_HWNDPREV
 NBWinConstants -> #GW_OWNER
 NBWinConstants -> #HEAP_CREATE_ENABLE_EXECUTE
 NBWinConstants -> #HEAP_GENERATE_EXCEPTIONS
 NBWinConstants -> #HEAP_NO_SERIALIZE
 NBWinConstants -> #HEAP_REALLOC_IN_PLACE_ONLY
 NBWinConstants -> #HEAP_ZERO_MEMORY
 NBWinConstants -> #HIGH_PRIORITY_CLASS
 NBWinConstants -> #IDABORT
 NBWinConstants -> #IDCANCEL
 NBWinConstants -> #IDCONTINUE
 NBWinConstants -> #IDIGNORE
 NBWinConstants -> #IDLE_PRIORITY_CLASS
 NBWinConstants -> #IDNO
 NBWinConstants -> #IDOK
 NBWinConstants -> #IDRETRY
 NBWinConstants -> #IDTRYAGAIN
 NBWinConstants -> #IDYES
 NBWinConstants -> #MB_ABORTRETRYIGNORE
 NBWinConstants -> #MB_APPLMODAL
 NBWinConstants -> #MB_CANCELTRYCONTINUE
 NBWinConstants -> #MB_DEFAULT_DESKTOP_ONLY
 NBWinConstants -> #MB_DEFBUTTON1
 NBWinConstants -> #MB_DEFBUTTON2
 NBWinConstants -> #MB_DEFBUTTON3
 NBWinConstants -> #MB_DEFBUTTON4
 NBWinConstants -> #MB_HELP
 NBWinConstants -> #MB_ICONASTERISK
 NBWinConstants -> #MB_ICONERROR
 NBWinConstants -> #MB_ICONEXCLAMATION
 NBWinConstants -> #MB_ICONHAND
 NBWinConstants -> #MB_ICONINFORMATION

NBWinConstants -> #MB_ICONQUESTION
 NBWinConstants -> #MB_ICONSTOP
 NBWinConstants -> #MB_ICONWARNING
 NBWinConstants -> #MB_OK
 NBWinConstants -> #MB_OKCANCEL
 NBWinConstants -> #MB_RETRYCANCEL
 NBWinConstants -> #MB_RIGHT
 NBWinConstants -> #MB_RTLCHECKING
 NBWinConstants -> #MB_SERVICE_NOTIFICATION
 NBWinConstants -> #MB_SETFOREGROUND
 NBWinConstants -> #MB_SYSTEMMODAL
 NBWinConstants -> #MB_TASKMODAL
 NBWinConstants -> #MB_TOPMOST
 NBWinConstants -> #MB_YESNO
 NBWinConstants -> #MB_YESNOCANCEL
 NBWinConstants -> #NORMAL_PRIORITY_CLASS
 NBWinConstants -> #PFD_DEPTH_DONTCARE
 NBWinConstants -> #PFD_DOUBLEBUFFER
 NBWinConstants -> #PFD_DOUBLEBUFFER_DONTCARE
 NBWinConstants -> #PFD_DRAW_TO_BITMAP
 NBWinConstants -> #PFD_DRAW_TO_WINDOW
 NBWinConstants -> #PFD_GENERIC_ACCELERATED
 NBWinConstants -> #PFD_GENERIC_FORMAT
 NBWinConstants -> #PFD_MAIN_PLANE
 NBWinConstants -> #PFD_NEED_PALETTE
 NBWinConstants -> #PFD_NEED_SYSTEM_PALETTE
 NBWinConstants -> #PFD_OVERLAY_PLANE
 NBWinConstants -> #PFD_STEREO
 NBWinConstants -> #PFD_STEREO_DONTCARE
 NBWinConstants -> #PFD_SUPPORT_DIRECTDRAW
 NBWinConstants -> #PFD_SUPPORT_GDI
 NBWinConstants -> #PFD_SUPPORT_OPENGL
 NBWinConstants -> #PFD_SWAP_COPY
 NBWinConstants -> #PFD_SWAP_EXCHANGE
 NBWinConstants -> #PFD_SWAP_LAYER_BUFFERS
 NBWinConstants -> #PFD_TYPE_COLORINDEX
 NBWinConstants -> #PFD_TYPE_RGBA
 NBWinConstants -> #PFD_UNDERLAY_PLANE
 NBWinConstants -> #REALTIME_PRIORITY_CLASS
 NBWinConstants -> #SM_ARRANGE
 NBWinConstants -> #SM_CLEANBOOT
 NBWinConstants -> #SM_MONITORS
 NBWinConstants -> #SM_CMOUSEBUTTONS
 NBWinConstants -> #SM_CXBORDER
 NBWinConstants -> #SM_CXCURSOR
 NBWinConstants -> #SM_CXDLGFRAME
 NBWinConstants -> #SM_CXDOUBLECLK
 NBWinConstants -> #SM_CXDRAG
 NBWinConstants -> #SM_CXEDGE
 NBWinConstants -> #SM_CXFIXEDFRAME
 NBWinConstants -> #SM_CXFOCUSBORDER
 NBWinConstants -> #SM_CXFRAME
 NBWinConstants -> #SM_CXFULLSCREEN
 NBWinConstants -> #SM_CXHSCROLL
 NBWinConstants -> #SM_CXHTHUMB
 NBWinConstants -> #SM_CXICON
 NBWinConstants -> #SM_CXICONSPACING
 NBWinConstants -> #SM_CXMAXIMIZED
 NBWinConstants -> #SM_CXMAXTRACK

NBWinConstants -> #SM_CXMENUCHECK
 NBWinConstants -> #SM_CXMENUSIZE
 NBWinConstants -> #SM_CXMIN
 NBWinConstants -> #SM_CXMINIMIZED
 NBWinConstants -> #SM_CXMINSPACING
 NBWinConstants -> #SM_CXMINTRACK
 NBWinConstants -> #SM_CXPADDEDBORDER
 NBWinConstants -> #SM_CXSCREEN
 NBWinConstants -> #SM_CXSIZE
 NBWinConstants -> #SM_CXSIZEFRAME
 NBWinConstants -> #SM_CXSMICON
 NBWinConstants -> #SM_CXSMSIZE
 NBWinConstants -> #SM_CXVIRTUALSCREEN
 NBWinConstants -> #SM_CXVSCROLL
 NBWinConstants -> #SM_CYBORDER
 NBWinConstants -> #SM_CYCAPTION
 NBWinConstants -> #SM_CYCURSOR
 NBWinConstants -> #SM_CYDLGFRAME
 NBWinConstants -> #SM_CYDOUBLECLK
 NBWinConstants -> #SM_CYDRAG
 NBWinConstants -> #SM_CYEDGE
 NBWinConstants -> #SM_CYFIXEDFRAME
 NBWinConstants -> #SM_CYFOCUSBORDER
 NBWinConstants -> #SM_CYFRAME
 NBWinConstants -> #SM_CYFULLSCREEN
 NBWinConstants -> #SM_CYHSCROLL
 NBWinConstants -> #SM_CYICON
 NBWinConstants -> #SM_CYICONSPACING
 NBWinConstants -> #SM_CYKANJIWINDOW
 NBWinConstants -> #SM_CYMAXIMIZED
 NBWinConstants -> #SM_CYMAXTRACK
 NBWinConstants -> #SM_CYMENU
 NBWinConstants -> #SM_CYMENUCHECK
 NBWinConstants -> #SM_CYMENUSIZE
 NBWinConstants -> #SM_CYMIN
 NBWinConstants -> #SM_CYMINIMIZED
 NBWinConstants -> #SM_CYMINSPACING
 NBWinConstants -> #SM_CYMINTRACK
 NBWinConstants -> #SM_CYSCREEN
 NBWinConstants -> #SM_CYSIZE
 NBWinConstants -> #SM_CYSIZEFRAME
 NBWinConstants -> #SM_CYSMCAPTION
 NBWinConstants -> #SM_CYSMICON
 NBWinConstants -> #SM_CYSMSIZE
 NBWinConstants -> #SM_CYVIRTUALSCREEN
 NBWinConstants -> #SM_CYVSCROLL
 NBWinConstants -> #SM_CYVTHUMB
 NBWinConstants -> #SM_DBCSENABLED
 NBWinConstants -> #SM_DEBUG
 NBWinConstants -> #SM_DIGITIZER
 NBWinConstants -> #SM_IMMENABLED
 NBWinConstants -> #SM_MAXIMUMTOUCHES
 NBWinConstants -> #SM_MEDIACENTER
 NBWinConstants -> #SM_MENUDROPALIGNMENT
 NBWinConstants -> #SM_MIDEASTENABLED
 NBWinConstants -> #SM_MOUSEHORIZONTALWHEELPRESENT
 NBWinConstants -> #SM_MOUSEPRESENT
 NBWinConstants -> #SM_MOUSEWHEELPRESENT
 NBWinConstants -> #SM_NETWORK
 NBWinConstants -> #SM_PENWINDOWS
 NBWinConstants -> #SM_REMOTECONTROL
 NBWinConstants -> #SM_REMOTESESSION
 NBWinConstants -> #SM_SAMEDISPLAYFORMAT
 NBWinConstants -> #SM_SECURE
 NBWinConstants -> #SM_SHOWSOUNDS
 NBWinConstants -> #SM_SHUTTINGDOWN
 NBWinConstants -> #SM_SLOWMACHINE
 NBWinConstants -> #SM_STARTER
 NBWinConstants -> #SM_SWAPBUTTON
 NBWinConstants -> #SM_TABLETPC
 NBWinConstants -> #SM_XVIRTUALSCREEN
 NBWinConstants -> #SM_YVIRTUALSCREEN
 NBWinConstants -> #SPECIFIC_RIGHTS_ALL
 NBWinConstants -> #STANDARD_RIGHTS_ALL
 NBWinConstants -> #STANDARD_RIGHTS_EXECUTE
 NBWinConstants -> #STANDARD_RIGHTS_READ
 NBWinConstants -> #STANDARD_RIGHTS_REQUIRED
 NBWinConstants -> #STANDARD_RIGHTS_WRITE
 NBWinConstants -> #SW_FORCEMINIMIZE
 NBWinConstants -> #SW_HIDE
 NBWinConstants -> #SW_MAX
 NBWinConstants -> #SW_MAXIMIZE
 NBWinConstants -> #SW_MINIMIZE
 NBWinConstants -> #SW_NORMAL
 NBWinConstants -> #SW_RESTORE
 NBWinConstants -> #SW_SHOW
 NBWinConstants -> #SW_SHOWDEFAULT
 NBWinConstants -> #SW_SHOWMAXIMIZED
 NBWinConstants -> #SW_SHOWMINIMIZED
 NBWinConstants -> #SW_SHOWMINNOACTIVE
 NBWinConstants -> #SW_SHOWNA
 NBWinConstants -> #SW_SHOWNOACTIVATE
 NBWinConstants -> #SW_SHOWNORMAL
 NBWinConstants -> #SYNCHRONIZE
 NBWinConstants -> #THREAD_ALL_ACCESS
 NBWinConstants -> #THREAD_DIRECT_IMPERSONATION
 NBWinConstants -> #THREAD_GET_CONTEXT
 NBWinConstants -> #THREAD_IMPERSONATE
 NBWinConstants -> #THREAD_QUERY_INFORMATION
 NBWinConstants -> #THREAD_SET_CONTEXT
 NBWinConstants -> #THREAD_SET_INFORMATION
 NBWinConstants -> #THREAD_SET_THREAD_TOKEN
 NBWinConstants -> #THREAD_SUSPEND_RESUME
 NBWinConstants -> #THREAD_TERMINATE
 NBWinConstants -> #WM_ACTIVATEAPP
 NBWinConstants -> #WM_CANCELMODE
 NBWinConstants -> #WM_CHILDACTIVATE
 NBWinConstants -> #WM_CLOSE
 NBWinConstants -> #WM_COMPACTING
 NBWinConstants -> #WM_CREATE
 NBWinConstants -> #WM_DESTROY
 NBWinConstants -> #WM_ENABLE
 NBWinConstants -> #WM_ENTERSIZEMOVE
 NBWinConstants -> #WM_EXITSIZEMOVE
 NBWinConstants -> #WM_GETICON
 NBWinConstants -> #WM_GETMINMAXINFO
 NBWinConstants -> #WM_INPUTLANGCHANGE
 NBWinConstants -> #WM_INPUTLANGCHANGEREQUEST
 NBWinConstants -> #WM_MOVE

NBWinConstants -> #WM_MOVING
 NBWinConstants -> #WM_NCACTIVATE
 NBWinConstants -> #WM_NCCALCSIZE
 NBWinConstants -> #WM_NCCREATE
 NBWinConstants -> #WM_NCDestroy
 NBWinConstants -> #WM_NULL
 NBWinConstants -> #WM_QUERYDRAGICON
 NBWinConstants -> #WM_QUERYOPEN
 NBWinConstants -> #WM_QUIT
 NBWinConstants -> #WM_SHOWWINDOW
 NBWinConstants -> #WM_SIZE
 NBWinConstants -> #WM_SIZING
 NBWinConstants -> #WM_STYLECHANGED
 NBWinConstants -> #WM_STYLECHANGING
 NBWinConstants -> #WM_THEMECHANGED
 NBWinConstants -> #WM_USERCHANGED
 NBWinConstants -> #WM_WINDOWPOSCHANGED
 NBWinConstants -> #WM_WINDOWPOSCHANGING
 NBWinConstants -> #WS_BORDER
 NBWinConstants -> #WS_CAPTION
 NBWinConstants -> #WS_CHILD
 NBWinConstants -> #WS_CHILDWINDOW
 NBWinConstants -> #WS_CLIPCHILDREN
 NBWinConstants -> #WS_CLIPSIBLINGS
 NBWinConstants -> #WS_DISABLED
 NBWinConstants -> #WS_DLGFRAME
 NBWinConstants -> #WS_EX_ACCEPTFILES
 NBWinConstants -> #WS_EX_APPWINDOW
 NBWinConstants -> #WS_EX_CLIENTEDGE
 NBWinConstants -> #WS_EX_COMPOSITED
 NBWinConstants -> #WS_EX_CONTEXTHELP
 NBWinConstants -> #WS_EX_CONTROLparent
 NBWinConstants -> #WS_EX_DLGMODALFRAME
 NBWinConstants -> #WS_EX_LAYERED
 NBWinConstants -> #WS_EX_LAYOUTRTL
 NBWinConstants -> #WS_EX_LEFT
 NBWinConstants -> #WS_EX_LEFTSCROLLBAR
 NBWinConstants -> #WS_EX_LTRREADING
 NBWinConstants -> #WS_EX_MDICHILD
 NBWinConstants -> #WS_EX_NOACTIVATE
 NBWinConstants -> #WS_EX_NOINHERITLAYOUT
 NBWinConstants -> #WS_EX_NOPARENTNOTIFY
 NBWinConstants -> #WS_EX_OVERLAPPEDWINDOW
 NBWinConstants -> #WS_EX_PALETTEWINDOW
 NBWinConstants -> #WS_EX_RIGHT
 NBWinConstants -> #WS_EX_RIGHTSCROLLBAR
 NBWinConstants -> #WS_EX_RTLREADING
 NBWinConstants -> #WS_EX_STATICEDGE
 NBWinConstants -> #WS_EX_TOOLWINDOW
 NBWinConstants -> #WS_EX_TOPMOST
 NBWinConstants -> #WS_EX_TRANSPARENT
 NBWinConstants -> #WS_EX_WINDOWEDGE
 NBWinConstants -> #WS_GROUP
 NBWinConstants -> #WS_HSCROLL
 NBWinConstants -> #WS_ICONIC
 NBWinConstants -> #WS_MAXIMIZE
 NBWinConstants -> #WS_MAXIMIZEBOX
 NBWinConstants -> #WS_MINIMIZE
 NBWinConstants -> #WS_MINIMIZEBOX
 NBWinConstants -> #WS_OVERLAPPED
 NBWinConstants -> #WS_OVERLAPPEDWINDOW
 NBWinConstants -> #WS_POPUP
 NBWinConstants -> #WS_POPUPWINDOW
 NBWinConstants -> #WS_SIZEBOX
 NBWinConstants -> #WS_SYSMENU
 NBWinConstants -> #WS_TABSTOP
 NBWinConstants -> #WS_THICKFRAME
 NBWinConstants -> #WS_TILED
 NBWinConstants -> #WS_TILEDWINDOW
 NBWinConstants -> #WS_VISIBLE
 NBWinConstants -> #WS_VSCROLL
 NBWinTypes -> #ATOM
 NBWinTypes -> #BOOL
 NBWinTypes -> #BOOLEAN
 NBWinTypes -> #BYTE
 NBWinTypes -> #CALLBACK
 NBWinTypes -> #CHAR
 NBWinTypes -> #COLORREF
 NBWinTypes -> #DWORD
 NBWinTypes -> #DWORD32
 NBWinTypes -> #DWORD64
 NBWinTypes -> #DWORDLONG
 NBWinTypes -> #DWORD_PTR
 NBWinTypes -> #FLOAT
 NBWinTypes -> #HACCEL
 NBWinTypes -> #HALF_PTR
 NBWinTypes -> #HANDLE
 NBWinTypes -> #HBRUSH
 NBWinTypes -> #HCOLORSPACE
 NBWinTypes -> #HCONV
 NBWinTypes -> #HCONVLIST
 NBWinTypes -> #HCURSOR
 NBWinTypes -> #HDC
 NBWinTypes -> #HDDDEDATA
 NBWinTypes -> #HDESK
 NBWinTypes -> #HDROP
 NBWinTypes -> #HDWP
 NBWinTypes -> #HENHMETAFILE
 NBWinTypes -> #HFILE
 NBWinTypes -> #HFONT
 NBWinTypes -> #HGDIOBJ
 NBWinTypes -> #HGLOBAL
 NBWinTypes -> #HHOOK
 NBWinTypes -> #HICON
 NBWinTypes -> #HINSTANCE
 NBWinTypes -> #HKEY
 NBWinTypes -> #HKL
 NBWinTypes -> #HLOCAL
 NBWinTypes -> #HMENU
 NBWinTypes -> #HMETAFILE
 NBWinTypes -> #HMODULE
 NBWinTypes -> #HMONITOR
 NBWinTypes -> #HPALETTE
 NBWinTypes -> #HPEN
 NBWinTypes -> #HRESULT
 NBWinTypes -> #HRGN
 NBWinTypes -> #HRSRC
 NBWinTypes -> #HSZ
 NBWinTypes -> #HWINSTA

```

NBWinTypes -> #HWND
NBWinTypes -> #INT
NBWinTypes -> #INT32
NBWinTypes -> #INT64
NBWinTypes -> #INT_PTR
NBWinTypes -> #LANGID
NBWinTypes -> #LCID
NBWinTypes -> #LCTYPE
NBWinTypes -> #LGRPID
NBWinTypes -> #LONG
NBWinTypes -> #LONG32
NBWinTypes -> #LONG64
NBWinTypes -> #LONGLONG
NBWinTypes -> #LONG_PTR
NBWinTypes -> #LPARAM
NBWinTypes -> #LPBOOL
NBWinTypes -> #LPBYTE
NBWinTypes -> #LPCOLORREF
NBWinTypes -> #LPCSTR
NBWinTypes -> #LPCTSTR
NBWinTypes -> #LPCVOID
NBWinTypes -> #LPCWSTR
NBWinTypes -> #LPDWORD
NBWinTypes -> #LPHANDLE
NBWinTypes -> #LPINT
NBWinTypes -> #LPLONG
NBWinTypes -> #LPSTR
NBWinTypes -> #LPTCH
NBWinTypes -> #LPTSTR
NBWinTypes -> #LPVOID
NBWinTypes -> #LPWCH
NBWinTypes -> #LPWORD
NBWinTypes -> #LPWSTR
NBWinTypes -> #LRESULT
NBWinTypes -> #PBOOL
NBWinTypes -> #PBOOLEAN
NBWinTypes -> #PBYTE
NBWinTypes -> #PCHAR
NBWinTypes -> #PCSTR
NBWinTypes -> #PCTSTR
NBWinTypes -> #PCWSTR
NBWinTypes -> #PDWORD
NBWinTypes -> #PDWORD32
NBWinTypes -> #PDWORD64
NBWinTypes -> #PDWORDLONG
NBWinTypes -> #PDWORD_PTR
NBWinTypes -> #PFLOAT
NBWinTypes -> #PHALF_PTR
NBWinTypes -> #PHANDLE
NBWinTypes -> #PHKEY
NBWinTypes -> #PINT
NBWinTypes -> #PINT32
NBWinTypes -> #PINT64
NBWinTypes -> #PINT_PTR
NBWinTypes -> #PLCID
NBWinTypes -> #PLONG
NBWinTypes -> #PLONG32
NBWinTypes -> #PLONG64
NBWinTypes -> #PLONGLONG
NBWinTypes -> #PLONG_PTR
NBWinTypes -> #POINT
NBWinTypes -> #POINTER_32
NBWinTypes -> #POINTER_64
NBWinTypes -> #PSHORT
NBWinTypes -> #PSIZE_T
NBWinTypes -> #PSSIZE_T
NBWinTypes -> #PSTR
NBWinTypes -> #PTBYTE
NBWinTypes -> #PTCHAR
NBWinTypes -> #PTSTR
NBWinTypes -> #PUCHAR
NBWinTypes -> #PUHALF_PTR
NBWinTypes -> #PUINT
NBWinTypes -> #PUINT32
NBWinTypes -> #PUINT64
NBWinTypes -> #PUINT_PTR
NBWinTypes -> #PULONG
NBWinTypes -> #PULONG32
NBWinTypes -> #PULONG64
NBWinTypes -> #PULONGLONG
NBWinTypes -> #PULONG_PTR
NBWinTypes -> #PUSHORT
NBWinTypes -> #PVOID
NBWinTypes -> #PWCHAR
NBWinTypes -> #PWORD
NBWinTypes -> #PWSTR
NBWinTypes -> #RECT
NBWinTypes -> #SC_HANDLE
NBWinTypes -> #SC_LOCK
NBWinTypes -> #SERVICE_STATUS_HANDLE
NBWinTypes -> #SHORT
NBWinTypes -> #SIZE_T
NBWinTypes -> #SSIZE_T
NBWinTypes -> #TBYTE
NBWinTypes -> #TCHAR
NBWinTypes -> #UCHAR
NBWinTypes -> #UHALF_PTR
NBWinTypes -> #UINT
NBWinTypes -> #UINT32
NBWinTypes -> #UINT64
NBWinTypes -> #UINT_PTR
NBWinTypes -> #ULONG
NBWinTypes -> #ULONG32
NBWinTypes -> #ULONG64
NBWinTypes -> #ULONGLONG
NBWinTypes -> #ULONG_PTR
NBWinTypes -> #USHORT
NBWinTypes -> #USN
NBWinTypes -> #VOID
NBWinTypes -> #WCHAR
NBWinTypes -> #WNDCLASSEX
NBWinTypes -> #WNDPROC
NBWinTypes -> #WORD
NBWinTypes -> #WPARAM
NativeBoostConstants -> #ErrInvalidPlatformId
NativeBoostConstants -> #ErrInvalidPrimitive VoltageUse
NativeBoostConstants -> #ErrNoNBPrimitive
NativeBoostConstants -> #ErrNoNativeCodeInMethod
NativeBoostConstants -> #ErrNotEnabled

```


NativeBoostConstants -> #ErrRunningViaInterpreter
 NativeBoostConstants -> #Linux32PlatformId
 NativeBoostConstants -> #Mac32PlatformId
 NativeBoostConstants -> #NBErrorBase
 NativeBoostConstants -> #NBErrorDescriptions
 NativeBoostConstants -> #NBPrimErrBadArgument
 NativeBoostConstants -> #NBPrimErrBadIndex
 NativeBoostConstants -> #NBPrimErrBadMethod
 NativeBoostConstants -> #NBPrimErrBadNumArgs
 NativeBoostConstants -> #NBPrimErrBadReceiver
 NativeBoostConstants -> #NBPrimErrGenericFailure
 NativeBoostConstants -> #NBPrimErrInappropriate
 NativeBoostConstants -> #NBPrimErrLimitExceeded
 NativeBoostConstants -> #NBPrimErrNamedInternal
 NativeBoostConstants -> #NBPrimErrNoCMemory
 NativeBoostConstants -> #NBPrimErrNoMemory
 NativeBoostConstants -> #NBPrimErrNoModification
 NativeBoostConstants -> #NBPrimErrNotFound
 NativeBoostConstants -> #NBPrimErrObjectMayMove
 NativeBoostConstants -> #NBPrimErrUnsupported
 NativeBoostConstants -> #NBPrimNoErr
 NativeBoostConstants -> #Win32PlatformId
 NetNameResolver -> #ResolverBusy
 NetNameResolver -> #ResolverError
 NetNameResolver -> #ResolverMutex
 NetNameResolver -> #ResolverReady
 NetNameResolver -> #ResolverUninitialized
 OCASTTranslator -> #OptimizedMessages
 PNGReadWriter -> #BPP
 PNGReadWriter -> #BlockHeight
 PNGReadWriter -> #BlockWidth
 PNGReadWriter -> #StandardColors
 PNGReadWriter -> #StandardSwizzleMaps
 ParseNode -> #Bfp
 ParseNode -> #BtpLong
 ParseNode -> #CodeBases
 ParseNode -> #CodeLimits
 ParseNode -> #DbExtDoAll
 ParseNode -> #Dup
 ParseNode -> #EndMethod
 ParseNode -> #EndRemote
 ParseNode -> #Jmp
 ParseNode -> #JmpLimit
 ParseNode -> #JmpLong
 ParseNode -> #LdFalse
 ParseNode -> #LdInstLong
 ParseNode -> #LdInstType
 ParseNode -> #LdLitIndType
 ParseNode -> #LdLitType
 ParseNode -> #LdMinus1
 ParseNode -> #LdNil
 ParseNode -> #LdSelf
 ParseNode -> #LdSuper
 ParseNode -> #LdTempType
 ParseNode -> #LdThisContext
 ParseNode -> #LdTrue
 ParseNode -> #LoadLong
 ParseNode -> #LongLongDoAll
 ParseNode -> #NodeFalse
 ParseNode -> #NodeNil
 ParseNode -> #NodeSelf
 ParseNode -> #NodeSuper
 ParseNode -> #NodeThisContext
 ParseNode -> #NodeTrue
 ParseNode -> #Pop
 ParseNode -> #Send
 ParseNode -> #SendLimit
 ParseNode -> #SendLong
 ParseNode -> #SendLong2
 ParseNode -> #SendPlus
 ParseNode -> #SendType
 ParseNode -> #ShortStoP
 ParseNode -> #StdLiterals
 ParseNode -> #StdSelectors
 ParseNode -> #StdVariables
 ParseNode -> #Store
 ParseNode -> #StorePop
 ProcessorScheduler -> #HighIOPriority
 ProcessorScheduler -> #LowIOPriority
 ProcessorScheduler -> #SystemBackgroundPriority
 ProcessorScheduler -> #SystemRockBottomPriority
 ProcessorScheduler -> #TimingPriority
 ProcessorScheduler -> #UserBackgroundPriority
 ProcessorScheduler -> #UserInterruptPriority
 ProcessorScheduler -> #UserSchedulingPriority
 RBAbstractClass -> #LookupSuperclass
 RBClass -> #LookupComment
 RBScanner -> #PatternVariableCharacter
 RBScanner -> #PatternVariableCharacter
 RBScanner -> #classificationTable
 RBTransformationRule -> #RecursiveSelfRule
 RealEstateAgent -> #StaggerOffset
 RemotesManager -> #addRemoteIcon
 RemotesManager -> #editRemoteIcon
 RemotesManager -> #removeRemoteIcon
 RxMatcher -> #Cr
 RxMatcher -> #Lf
 RxParser -> #BackslashConstants
 RxParser -> #BackslashSpecials
 RxsPredicate -> #EscapedLetterSelectors
 RxsPredicate -> #NamedClassSelectors
 SHA1 -> #K1
 SHA1 -> #K2
 SHA1 -> #K3
 SHA1 -> #K4
 Scanner -> #DoItCharacter
 SetElement -> #NilElement
 Socket -> #Connected
 Socket -> #DeadServer
 Socket -> #InvalidSocket
 Socket -> #OtherEndClosed
 Socket -> #TCPSocketType
 Socket -> #ThisEndClosed
 Socket -> #UDPSocketType
 Socket -> #Unconnected
 Socket -> #WaitingForConnection
 String -> #AsciiOrder
 String -> #CSLineEnders
 String -> #CSNonSeparators

String -> #CSSeparators
 String -> #CaseInsensitiveOrder
 String -> #CaseSensitiveOrder
 String -> #CrLfExchangeTable
 String -> #LowercasingTable
 String -> #Tokenish
 String -> #TypeTable
 String -> #UppercasingTable
 TextConstants -> #BS
 TextConstants -> #BS2
 TextConstants -> #Basal
 TextConstants -> #Bold
 TextConstants -> #CR
 TextConstants -> #Centered
 TextConstants -> #Clear
 TextConstants -> #CrossedX
 TextConstants -> #CtrlA
 TextConstants -> #CtrlB
 TextConstants -> #CtrlC
 TextConstants -> #CtrlD
 TextConstants -> #CtrlDigits
 TextConstants -> #CtrlE
 TextConstants -> #CtrlF
 TextConstants -> #CtrlG
 TextConstants -> #CtrlH
 TextConstants -> #CtrlI
 TextConstants -> #CtrlJ
 TextConstants -> #CtrlK
 TextConstants -> #CtrlL
 TextConstants -> #CtrlM
 TextConstants -> #CtrlN
 TextConstants -> #CtrlO
 TextConstants -> #CtrlOpenBrackets
 TextConstants -> #CtrlP
 TextConstants -> #CtrlQ
 TextConstants -> #CtrlR
 TextConstants -> #CtrlS
 TextConstants -> #CtrlT
 TextConstants -> #CtrlU
 TextConstants -> #CtrlV
 TextConstants -> #CtrlW
 TextConstants -> #CtrlX
 TextConstants -> #CtrlY
 TextConstants -> #CtrlZ
 TextConstants -> #Ctrla
 TextConstants -> #Ctrlb
 TextConstants -> #Ctrlc
 TextConstants -> #Ctrl d
 TextConstants -> #Ctrl e
 TextConstants -> #Ctrlf
 TextConstants -> #Ctrlg
 TextConstants -> #Ctrlh
 TextConstants -> #Ctrli
 TextConstants -> #Ctrlj
 TextConstants -> #Ctrlk
 TextConstants -> #Ctrl l
 TextConstants -> #Ctrlm
 TextConstants -> #Ctrl n
 TextConstants -> #Ctrl o
 TextConstants -> #Ctrlp

TextConstants -> #Ctrlq
 TextConstants -> #Ctrlr
 TextConstants -> #Ctrl s
 TextConstants -> #Ctrlt
 TextConstants -> #Ctrlu
 TextConstants -> #Ctrlv
 TextConstants -> #Ctrlw
 TextConstants -> #Ctrlx
 TextConstants -> #Ctrl y
 TextConstants -> #Ctrlz
 TextConstants -> #DefaultBaseline
 TextConstants -> #DefaultFontSize
 TextConstants -> #DefaultLineGrid
 TextConstants -> #DefaultMarginTabsArray
 TextConstants -> #DefaultMask
 TextConstants -> #DefaultRule
 TextConstants -> #DefaultSpace
 TextConstants -> #DefaultTab
 TextConstants -> #DefaultTabsArray
 TextConstants -> #ESC
 TextConstants -> #EndOfRun
 TextConstants -> #Enter
 TextConstants -> #Italic
 TextConstants -> #Justified
 TextConstants -> #LeftFlush
 TextConstants -> #LeftMarginTab
 TextConstants -> #RightFlush
 TextConstants -> #RightMarginTab
 TextConstants -> #Space
 TextConstants -> #Tab
 TextConstants -> #TextSharedInformation
 TextContainer -> #OuterMargin
 TextConverter -> #latin1Encodings
 TextConverter -> #latin1Map
 ThumbnailMorph -> #EccentricityThreshold
 ThumbnailMorph -> #RecursionMax
 Transcriber -> #Icon
 TransferMorph -> #CopyPlusIcon
 UCSTable -> #GB2312Table
 UCSTable -> #JISX0208Table
 UCSTable -> #KSX1001Table
 UCSTable -> #Latin1Table
 Unicode -> #Cc
 Unicode -> #Cf
 Unicode -> #Cn
 Unicode -> #Co
 Unicode -> #Cs
 Unicode -> #DecimalProperty
 Unicode -> #GeneralCategory
 Unicode -> #Ll
 Unicode -> #Lm
 Unicode -> #Lo
 Unicode -> #Lt
 Unicode -> #Lu
 Unicode -> #Mc
 Unicode -> #Me
 Unicode -> #Mn
 Unicode -> #Nd
 Unicode -> #Nl

Unicode -> #No
 Unicode -> #Pc
 Unicode -> #Pd
 Unicode -> #Pe
 Unicode -> #Pf
 Unicode -> #Pi
 Unicode -> #Po
 Unicode -> #Ps
 Unicode -> #Sc
 Unicode -> #Sk
 Unicode -> #Sm
 Unicode -> #So
 Unicode -> #ToCasefold
 Unicode -> #ToLower
 Unicode -> #ToUpper
 Unicode -> #Zl
 Unicode -> #Zp
 Unicode -> #Zs
 ZipConstants -> #BaseDistance
 ZipConstants -> #BaseLength
 ZipConstants -> #BitLengthOrder
 ZipConstants -> #DistanceCodes
 ZipConstants -> #DynamicBlock
 ZipConstants -> #EndBlock
 ZipConstants -> #ExtraBitLengthBits
 ZipConstants -> #ExtraDistanceBits
 ZipConstants -> #ExtraLengthBits
 ZipConstants -> #FixedBlock
 ZipConstants -> #FixedDistanceTree
 ZipConstants -> #FixedLiteralTree
 ZipConstants -> #HashBits
 ZipConstants -> #HashMask
 ZipConstants -> #HashShift
 ZipConstants -> #MatchLengthCodes
 ZipConstants -> #MaxBitLengthBits
 ZipConstants -> #MaxBitLengthCodes
 ZipConstants -> #MaxBits
 ZipConstants -> #MaxDistCodes
 ZipConstants -> #MaxDistance
 ZipConstants -> #MaxLengthCodes
 ZipConstants -> #MaxLiteralCodes
 ZipConstants -> #MaxMatch
 ZipConstants -> #MinMatch
 ZipConstants -> #NumLiterals
 ZipConstants -> #Repeat11To138
 ZipConstants -> #Repeat3To10
 ZipConstants -> #Repeat3To6
 ZipConstants -> #StoredBlock
 ZipConstants -> #WindowMask
 ZipConstants -> #WindowSize
 ZipFileConstants -> #CentralDirectoryFileHeaderSignature
 ZipFileConstants -> #CompressionDeflated
 ZipFileConstants -> #CompressionLevelDefault
 ZipFileConstants -> #CompressionLevelNone
 ZipFileConstants -> #CompressionStored
 ZipFileConstants -> #DataDescriptorLength
 ZipFileConstants -> #DefaultDirectoryPermissions
 ZipFileConstants -> #DefaultFilePermissions
 ZipFileConstants -> #DeflatingCompressionFast
 ZipFileConstants -> #DeflatingCompressionMaximum
 ZipFileConstants -> #DeflatingCompressionNormal
 ZipFileConstants -> #DeflatingCompressionSuperFast
 ZipFileConstants -> #DirectoryAttrib
 ZipFileConstants -> #EndOfCentralDirectorySignature
 ZipFileConstants -> #FaMsdos
 ZipFileConstants -> #FaUnix
 ZipFileConstants -> #FileAttrib
 ZipFileConstants -> #IfaBinaryFile
 ZipFileConstants -> #IfaTextFile
 ZipFileConstants -> #LocalFileHeaderSignature
 ZnBase64Encoder -> #DefaultAlphabet
 ZnBase64Encoder -> #DefaultInverse
 ZnByteEncoder -> #ByteTextConverters
 ZnConstants -> #HTTPStatusCodes
 ZnHeaders -> #CommonHeaders
 ZnMimeType -> #ExtensionsMap
 ZnUTF8Encoder -> #ByteASCIISet
 ZnUTF8Encoder -> #ByteUTF8Encoding