This is a roadmap to the code layout in snit.tcl.

Package Definition

\* package provide

\* ::snit:: namespace definition; exports Snit commands.

Major Variable Definitions (this includes a whole lot of code)

\* ::snit:: variable definitions:

\* reservedArgs

\* prettyStackTrace Not used currently

\* ::snit::typeTemplate Template code shared by all Snit types.

As the type definition is parsed, it

produced text that gets inserted into

this template; then the template is

evaluated as though it were sourced

from a normal .tcl file.

\* Type namespace definition

\* User's typevariable definitions

\* Commands for use in type code

\* alias installhull

\* alias install

\* alias typevariable

\* alias variable

\* alias mytypevar

\* alias typevarname Deprecated

\* alias myvar

\* alias varname Deprecated

\* alias myproc

\* alias codename Deprecated

\* alias mymethod

\* alias mytypemethod

\* alias from

\* Snit's internal variables

\* See dictionary.txt

\* Template Code -- Stuff that gets filled in.

\* proc Snit\_instanceVars Initializes instance variables

\* proc Snit\_typeconstructor

\* Default Procs -- Stuff that's sometimes replaced.

\* proc Snit\_constructor The default constructor

\* proc Snit\_destructor The default destructor (empty)

\* %COMPILEDDEFS%

\* Call the Type Constructor

\* ::snit::nominalTypeProc Template for the normal type proc.

\* ::snit::simpleTypeProc Template for the simple type proc.

This is used when "-hastypemethods no";

all it does is create instances.

\* ::snit::nominalInstanceProc Template for the body of the normal

instance proc. Supports method

caching, delegation, etc.

\* ::snit::simpleInstanceProc Template for the body of the simple

instance proc, used when

"-simpledispatch yes". Doesn't

support delegation, upvar,

hierarchical methods, or exotic return

types.

\* Snit compilation variables

\* compiler The name of the slave interpreter used

to "compile" type definitions

\* compile Array, accumulates results of

"compiling" type definitions

\* reservedwords List of names that can't be used as

macros. Basically, any command

defined before the first macro.

Compilation Commands

\* proc ::snit::Comp.Init

\* proc ::snit::Comp.Compile

\* proc ::snit::Comp.SaveOptionInfo

\* proc ::snit::Comp.Define

\* proc ::snit::Comp.statement.pragma

\* proc ::snit::Comp.statement.widgetclass

\* proc ::snit::Comp.statement.hulltype

\* proc ::snit::Comp.statement.constructor

\* proc ::snit::Comp.statement.destructor

\* proc ::snit::Comp.statement.option

\* proc ::snit::Comp.OptionNameIsValid

\* proc ::snit::Comp.statement.oncget

\* proc ::snit::Comp.statement.onconfigure

\* proc ::snit::Comp.statement.method

\* proc ::snit::Comp.CheckMethodName

\* proc ::snit::Comp.statement.typemethod

\* proc ::snit::Comp.statement.typeconstructor

\* proc ::snit::Comp.statement.proc

\* proc ::snit::Comp.statement.typevariable

\* proc ::snit::Comp.statement.variable

\* proc ::snit::Comp.statement.typecomponent

\* proc ::snit::Comp.DefineTypeComponent

\* proc ::snit::Comp.statement.component

\* proc ::snit::Comp.DefineComponent

\* proc ::snit::Comp.statement.delegate

\* proc ::snit::Comp.DelegatedTypemethod

\* proc ::snit::Comp.DelegatedMethod

\* proc ::snit::Comp.DelegatedOption

\* proc ::snit::Comp.statement.expose

Public Commands

\* proc ::snit::compile

\* proc ::snit::type

\* proc ::snit::widgetadaptor

\* proc ::snit::widget

\* proc ::snit::typemethod

\* proc ::snit::method

\* proc ::snit::macro

Utility Commands

\* proc ::snit::Expand

\* proc ::snit::Mappend

\* proc ::snit::CheckArgs

\* proc ::snit::Capitalize

Snit Runtime Library

The commands defined here are used by Snit-generated code at run-time

rather than compile time.

\* Object Creation

\*\* ::snit::RT.type.typemethod.create

\*\* ::snit::RT.widget.typemethod.create

\*\* ::snit::RT.MakeInstanceCommand

\*\* ::snit::RT.InstanceTrace

\*\* ::snit::RT.ConstructInstance

\*\* ::snit::RT.UniqueName

\*\* ::snit::RT.UniqueInstanceNamespace

\*\* ::snit::RT.OptionDbGet

\* Object Destruction

\*\* ::snit::RT.method.destroy

\*\* ::snit::RT.DestroyObject

\*\* ::snit::RT.RemoveInstanceTrace

\* Typecomponent Management and Typemethod Caching

\*\* ::snit::RT.TypecomponentTrace

\*\* ::snit::RT.UnknownTypemethod

\* Component Management and Method Caching

\*\* ::snit::RT.Component

\*\* ::snit::RT.ComponentTrace

\*\* ::snit::RT.UnknownMethod

\*\* ::snit::RT.ClearInstanceCaches

\* Component Installation

\*\* ::snit::RT.installhull

\*\* ::snit::RT.install

\* Method/Variable Name Qualification

\*\* ::snit::RT.variable

\*\* ::snit::RT.mytypevar

\*\* ::snit::RT.myvar

\*\* ::snit::RT.myproc

\*\* ::snit::RT.codename

\*\* ::snit::RT.mytypemethod

\*\* ::snit::RT.mymethod

\*\* ::snit::RT.CallInstance

\* Utilities

\*\* ::snit::RT.from

\* Type Destruction

\*\* ::snit::RT.typemethod.destroy

\* Option Handling

\*\* ::snit::RT.method.cget

\*\* ::snit::RT.CacheCgetCommand

\*\* ::snit::RT.method.configurelist

\*\* ::snit::RT.CacheConfigureCommand

\*\* ::snit::RT.method.configure

\*\* ::snit::RT.GetOptionDbSpec

\* Type Introspection

\*\* ::snit::RT.typemethod.info

\*\* ::snit::RT.typemethod.info.typevars

\*\* ::snit::RT.typemethod.info.typemethods

\*\* ::snit::RT.typemethod.info.instances

\* Instance Introspection

\*\* ::snit::RT.method.info

\*\* ::snit::RT.method.info.type

\*\* ::snit::RT.method.info.typevars

\*\* ::snit::RT.method.info.typemethods

\*\* ::snit::RT.method.info.methods

\*\* ::snit::RT.method.info.vars

\*\* ::snit::RT.method.info.options