ContextBuilder **ActorBasicBuilder** - systemBuilder : SystemBuilder? {val} = null # contextBuilder: ContextBuilder? {val} = null - contextName : String? = null # actorName : String? = null - contextAddress : String? = null # coroutineScope : CoroutineScope = DEFAULT SCOPE - contextProtocol : String? = null # discardMessage : Boolean = DEFAULT DISCARD MESSAGE - contextPort : Int? = null # confined : Boolean = DEFAULT CONFINED # ioBound : Boolean = DEFAULT IO BOUND - contextScope : CoroutineScope? = null - actors : MutableMap<String, TransientActorBasic>? = null # channelSize : Int = DEFAULT_CHANNEL_SIZE - params : MutableParameterMap = mutableParameterMap() # actorBody: TransientActorBody? = null actorBuilder : ActorBasicBuilder {val} ActorBasicBuilder(ContextBuilder?) = ActorBasicBuilder(this) + clear(): ActorBasicBuilder - actorBuilderFsm : ActorBasicBuilderFsm {val} = ActorBasicBuilderFsm(this) + addActorName(String) : ActorBasicBuilder ~ setActorScope(CoroutineScope) : ActorBasicBuilder ContextBuilder(SystemBuilder?) + addActorScope(CoroutineScope): ActorBasicBuilder + clear(): ContextBuilder + forcedAddActorScope(CoroutineScope): ActorBasicBuilder + addContextName(String) : ContextBuilder <<use>> + addGlobalScope(): ActorBasicBuilder + addContextAddress(String) : ContextBuilder + addContextProtocol(String) : ContextBuilder + addDiscardMessageOption(Boolean) : ActorBasicBuilder + addConfinedOption(Boolean) : ActorBasicBuilder + addContextPort(Int) : ContextBuilder + addloBoundOption(Boolean) : ActorBasicBuilder + addContextScope(CoroutineScope): ContextBuilder + addChannelSizeOption(Int): ActorBasicBuilder + addGlobalScope(): ContextBuilder + addActorBody(TransientActorBody): ActorBasicBuilder + addContextByAnnotation(QakContext) : ContextBuilder + addQActorBody(QActorBasic, + addParameter(String, Any) : ContextBuilder QActorBasic.(IApplMessage) -> Unit : ActorBasicBuilder + removeParameter(String) : ContextBuilder + addQActorMethodBody(Method, QActorBasic) : ActorBasicBuilder + withParameterMap(ParameterMap.() -> Unit): ContextBuilder + addAutoActorBody(Class<out AutoQActorBasic>, + newActorBasic() : ActorBasicBuilder AutoQActorBasic): ActorBasicBuilder + addAutoActorBody(Class<out AutoQActorBasic>): + newActorBasicFsm(): ActorBasicFsmBuilder ActorBasicBuilder + build(): TransientContext + addActorBasicClassBody(Class<out ActorBasic>): + buildInSystem(): Pair<TransientContext, SystemBuilder> ActorBasicBuilder ~ addActor(TransientActorBasic) : ContextBuilder + build(): TransientActorBasic + buildInContext(): Pair<TransientActorBasic, ActorBasicBuilder> + upgradeToFsmBuilder(): ActorBasicFsmBuilder <<use>> <<use>> + asFsmBuilder(): ActorBasicFsmBuilder 0..1 **SystemBuilder** - contextBuilder : ContextBuilder {val} **ActorBasicFsmBuilder** = ContextBuilder(this) - initalState : String? = null - hostName : String? = null - states : MutableMap<String, TransientState>? = null - scope : CoroutineScope = null - gActor: QActorBasicFsm? = null - contexts : MutableSet<TransientContext> {val} = mutableSetOf() - stateBuilder : StateBuilder? = null - params : MutableParameterMap = mutableParameterMap() - alreadyBuilt : LateSingleInit<T> {val} = lateSingleInit() ActorBasicFsmBuilder(ContextBuilder?) - builtEvent : MutableStateFlow<String> {val} = + clear(): ActorBasicBuilder MutableStateFlow("") + newState(): StateBuilder + addQActorBasicFsm(QActorBasicFsm) : ActorBasicFsmBuilder + addHostName(String) : SystemBuilder + setInitialState(String) ActorBasicFsmBuilder addContext(TransientContext) : SystemBuilder + newContext(): ContextBuilder + addActorFsmBody(TransientActorBasicFsmBody): ActorBasicFsmBuilder + addParameter(String, Any) : SystemBuilder + addAutoActorFsmBody(Class<out AutoQActorBasicFsm): + removeParameter(String) : SystemBuilder ActorBasicFsmBuilder + withParameterMap(ParameterMap.() -> Unit) SystemBuilder + build(): TransientActorBasicFsm + addScope(CoroutineScope) : SystemBuilder + buildInContext(): Pair<TransientActorBasicFsm, ActorBasicBuilder> + addGlobalScope() : SystemBuilder ~ addState(TransientState): ActorBasicFsmBuilder + waitBuild() {suspend} + addActorBody(TransientActorBasicBody): ActorBasicFsmBuilder + build(): TransientSystem {suspen}