Document

Project: Generic Object Pool Lib
Title: Class Diagram
Doc version: 0.4.1(May 24, 2020)
(Please refer to class_diagram_release_note.md file for version history)
Repo: https://github.com/911992/WAsys_simple_generic_object_pool
Author(s):
911992

Meta Legends

X On-page reference

Note.. Note

Authors

911992[principal]
https://github.com/911992

wasys::lib::generic_object_pool

«entity»
Generic_Object_Pool_Policy

+DEF_MAX_OBJ_COUNT:int := 64 {final}
+DEF_MIN_OBJ_COUNT:int := 0 {final}
+DEF_OBJ_CREATION_POLICY:Full_Pool_Object_Creation_Policy := Create_New_No_Pooling {final}
+DEF_INS:Generic_Object_Pool_Policy=new Generic_Object_Pool_Policy(DEF_MAX_OBJ_COUNT,DEF_MIN_OBJ_COUNT,DEF_OBJ_CREATION_POLICY) {final}
+min_object_count:int {final}
+max_object_count:int {public-readonly} {package-settable}
-full_pool_instancing_policy:Full_Pool_Object_Creation_Policy {final}

+Generic_Object_Pool_Policy(arg_min_obj_count:int,arg_max_obj_count:int,:Full_Pool_Object_Creation_Policy):Generic_Object_Pool_Policy

A:Poolable_Object

Object_Pool_Type_Wrapper
{leaf}

-pool:Object_Pool {final}

+Object_Pool_Type_Wrapper(:Object_Factory,
:Generic_Object_Pool_Policy,
arg_thread_safe:bool,
arg_register_pool:bool):Object_Pool_Type_Wrapper

A

«enum»
Full_Pool_Object_Creation_Policy

Return_Null
Create_New_No_Pooling
Create_New_Extend_Pool_Size
Wait_Till_One_Free

«creates»

«uses»

C

«interface»
Object_Pool

+get_an_instance(void):Poolable_Object
+release_an_instance(:Poolable_Object):void
+idle_objects_count(void):int
+available_objects_count(void):int
+working_object_count(void):int
+shutdown_pool(void):void
+pool_is_working(void):bool
+is_registered(void):bool
+get_policy(void):Generic_Object_Pool_Policy

B

D

«singleton(eager)»
Pool_Context
{leaf}

_-INSTANCE:Pool_Context = new Pool_Context()_ {final}
-ctx:Vector<Generic_Object_Pool> = new Vector(1,7) {final}

-Pool_Context(void):Pool_Context
+get_instance(void):Pool_Context
+get_pool_unregistered_synced(:Object_Factory,
:Generic_Object_Pool_Policy):Object_Pool {synchronized}
+get_pool_registered_synced(:Object_Factory,
:Generic_Object_Pool_Policy):Object_Pool {synchronized}
+get_pool(:Object_Factory,:Generic_Object_Pool_Policy,
arg_thread_safe:bool,
arg_register:bool):Object_Pool {synchronized}
+unregister_pool(:Object_Pool,arg_shutdown_pool:bool):void {synchronized}

C

E

«thread-safe»
Generic_Object_Pool_Safe_Guard

-pool:Object_Pool {final}

~Generic_Object_Pool_Safe(:Object_Pool,
):Generic_Object_Pool_Safe

«creates»

Lock on associated pool may be applied by some methods

«holds»

A

B

C

«interface»
java::lang
Runnable

«creates»

D

E

java::lang
AutoClosable

api

«pool»
Generic_Object_Pool

-factory:Object_Factory {final}
-policy:Generic_Object_Pool_Policy {final}
-pool:Vector<Poolable_Object>
-working_ins_count:int := 0
-pool_working:bool := true {volatile}
-notify_thread_run:Runnable := new Runnable() {final}
-null_run:Runnable := new Runnable() {final}
-release_obj_run:Runnable {final}
-registered:bool

~Generic_Object_Pool(:Object_Factory,
:Generic_Object_Pool_Policy):Generic_Object_Pool
-init_pool(void):void
~get_factory(void):Object_Factory
~set_as_registered(:bool):void
~get_policy(void):Generic_Object_Pool_Policy

«interface»
Poolable_Object

+post_create(void):void
+pre_destroy(void):void
+reset_state(void):void
+set_pool(:Object_Pool):void

«abstract»
Poolable_Object_Adapter

#pool:Object_Pool

+reset_state(void):void
+close():void{final}

«creates»

«holds»

«uses»

«interface»
Object_Factory

+create_object(void):Poolable_Object