

```
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_sim
ple_generic_object_pool
Author(s):
911992
```

Meta Legends

The diagram illustrates two types of meta elements:

- On-page reference:** Represented by a pink circle containing a black 'x'.
- Note:** Represented by a light blue rectangle with a folded corner, containing the text 'Note..'. The word 'Note' is written to the right of the rectangle.

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

```

classDiagram
    class 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
    }

    class Object_Pool_Type_Wrapper {
        -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
    }

    class Pool_Context {
        <<singleton(eager)>>
        -_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}
    }

    class Generic_Object_Pool {
        <<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
    }

    class Generic_Object_Pool_Safe_Guard {
        <<thread-safe>>
        -pool:Object_Pool {final}
        ~Generic_Object_Pool_Safe(:Object_Pool):Generic_Object_Pool_Safe
    }

    class Full_Pool_Object_Creation_Policy {
        <<enum>>
        Return_Null
        Create_New_No_Pooling
        Create_New_Extend_Pool_Size
        Wait_Till_One_Free
    }

    class Object_Pool {
        <<interface>>
        +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
    }

    class Poolable_Object {
        <<interface>>
        +post_create(void):void
        +pre_destroy(void):void
        +reset_state(void):void
        +set_pool(:Object_Pool):void
    }

    class Poolable_Object_Adapter {
        <<abstract>>
        #pool:Object_Pool
        +reset_state(void):void
        +close():void{final}
    }

    class Object_Factory {
        <<interface>>
        +create_object(void):Poolable_Object
    }

    Generic_Object_Pool_Policy --> Object_Pool_Type_Wrapper
    Object_Pool_Type_Wrapper --> Pool_Context
    Pool_Context --> Generic_Object_Pool
    Generic_Object_Pool --> Generic_Object_Pool_Safe_Guard
    Generic_Object_Pool_Safe_Guard --> Object_Pool
    Object_Pool --> Poolable_Object
    Poolable_Object --> Poolable_Object_Adapter
    Poolable_Object_Adapter --> Object_Factory
    Full_Pool_Object_Creation_Policy --> Object_Pool
    Object_Pool --> Generic_Object_Pool_Policy
    Object_Pool --> Generic_Object_Pool_Safe_Guard
    Object_Pool --> Poolable_Object
    Poolable_Object --> Poolable_Object_Adapter
    Poolable_Object_Adapter --> Object_Factory
    
```

The diagram illustrates the design of a Generic Object Pool. It includes the following components and relationships:

- Generic_Object_Pool_Policy** (Entity):
 - Attributes: `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).
 - Method: `Generic_Object_Pool_Policy(arg_min_obj_count: int, arg_max_obj_count: int, : Full_Pool_Object_Creation_Policy): Generic_Object_Pool_Policy`.
- Object_Pool_Type_Wrapper** (Leaf):
 - Attribute: `pool` (Object_Pool, final).
 - Method: `Object_Pool_Type_Wrapper(: Object_Factory, : Generic_Object_Pool_Policy, arg_thread_safe: bool, arg_register_pool: bool): Object_Pool_Type_Wrapper`.
- Pool_Context** (Leaf, Singleton):
 - Attributes: `_INSTANCE` (Pool_Context, new Pool_Context(), final), `ctx` (Vector<Generic_Object_Pool>, new Vector(1, 7), final).
 - Methods: `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}`.
- Generic_Object_Pool** (Pool):
 - Attributes: `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).
 - Methods: `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`.
- Generic_Object_Pool_Safe_Guard** (Thread-Safe):
 - Attribute: `pool` (Object_Pool, final).
 - Method: `Generic_Object_Pool_Safe(: Object_Pool): Generic_Object_Pool_Safe`.
- Full_Pool_Object_Creation_Policy** (Enum):
 - Values: `Return_Null`, `Create_New_No_Pooling`, `Create_New_Extend_Pool_Size`, `Wait_Till_One_Free`.
- Object_Pool** (Interface):
 - Methods: `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`.
- Poolable_Object** (Interface):
 - Methods: `post_create(void): void`, `pre_destroy(void): void`, `reset_state(void): void`, `set_pool(: Object_Pool): void`.
- Poolable_Object_Adapter** (Abstract Class):
 - Attributes: `pool` (Object_Pool).
 - Methods: `reset_state(void): void`, `close(): void {final}`.
- Object_Factory** (Interface):
 - Method: `create_object(void): Poolable_Object`.

Relationships:

- Generic_Object_Pool_Policy** creates **Object_Pool_Type_Wrapper**.
- Object_Pool_Type_Wrapper** uses **Pool_Context**.
- Pool_Context** holds **Generic_Object_Pool**.
- Generic_Object_Pool** creates **Generic_Object_Pool_Safe_Guard**.
- Generic_Object_Pool_Safe_Guard** implements **Object_Pool**.
- Object_Pool** implements **Poolable_Object**.
- Poolable_Object_Adapter** implements **Poolable_Object**.
- Object_Factory** implements **Object_Pool**.
- Full_Pool_Object_Creation_Policy** is associated with **Object_Pool**.
- Object_Pool** is associated with **Generic_Object_Pool_Policy**.
- Object_Pool** is associated with **Generic_Object_Pool_Safe_Guard**.
- Object_Pool** is associated with **Poolable_Object**.
- Poolable_Object_Adapter** is associated with **Object_Factory**.