

SRP_Stack



Added in 2.1.10

SRP Stacks are simple data structures useful for nested or recursive operations.

Method	Description
Clear	Removes all values from an SRP Stack.
Create	Creates an SRP Stack.
Count	Gets the number of elements in the stack.
Peek	Returns the value at the top of an SRP Stack without removing it.
Pop	Removes and returns the value at the top of an SRP Stack.
Push	Inserts a value at the top of an SRP Stack.
Release	Releases the handle to an SRP Stack.

Stack Defined

A stack is a last-in-first-out (LIFO) data structure. Values are pushed onto the top and popped off in reverse order.

When to Use SRP_Stack

Stacks are useful when you need to temporarily save some information before doing an operation and then restoring that information when you are done. A common use case in OpenInight is the `@Record`, `@Dict`, and `@ID` global system variables. If we use these variables without preserving their existing values, we risk destroying other routines using them. A common practice should be to save these onto a stack. Of course, there are routines built into OI for this specific case, but this is still a good way to demonstrate the usefulness of `SRP_Stack`.

```
// Put the system variables onto the stack
SRP_Stack("Push", Handle, @Record)
SRP_Stack("Push", Handle, @Dict)
SRP_Stack("Push", Handle, @ID)

// Now we can use them in our own code
Open "SomeTable" to hTable then
    Open "DICT.SomeTable" to @Dict then
        @ID = "SomeKey"
        Read @Record from hTable, @ID then
            // Do stuff
        end
    end
end

// Restore the system variables, but make sure to do it in the correct order
@ID = SRP_Stack("Pop", Handle)
@Dict = SRP_Stack("Pop", Handle)
@Record = SRP_Stack("Pop", Handle)
```

This, of course, is not the only use case. Anytime you have nested operations that use the same global variables, you'll want to use a stack to protect them.

SRP Stacks are SRP Lists

`SRP_Stack` is a wrapper around [SRP_List](#), so you can use an `SRP_Stack` handle with `SRP_List`. So, even though `SRP_Stack` doesn't have a `GetVariable` routine, you can use [SRP_List's](#).

Remember

Don't forget to release your SRP Stack handles. Always.