

# StackCore

Posted At : August 1, 2007 5:44 PM | Posted By : Jon Hartmann

Related Categories: ColdFusion, Experiments, Frameworks

I've come up with a .02 version of warpcore using a Java Stack as the event manager, but from simple testing it doesn't seem to be as fast as the previous version. I'm going to have to narrow this down to creation time or execution time though.

[Click more to see the code.](#)

```
<cfcomponent name="warpCore">

    <cffunction name="init" returntype="warpCore01" output="false">
        <cfargument name="trigger" type="string" required="false" default="request.func"/>

        <cfscript>
            variables.eventCore = CreateObject("java", "java.util.Stack");
            variables.driveCore = StructNew();
            variables.trigger = arguments.trigger;
            variables.cap = 10;
            return this;
        </cfscript>
    </cffunction>

    <cffunction name="addDrive">
        <cfargument name="controllerPath" type="string"/>

        <cfscript>
            var objController = CreateObject("component", arguments.controllerPath);
            var arrEvents = StructKeyArray(objController);
            var iEvents = ArrayLen(arrEvents);
            var i = 0;

            if (iEvents gt 0) {
                for(i = 1; i lte iEvents; i = i + 1) {
                    if (NOT StructKeyExists(variables.driveCore, arrEvents[i])) {
                        StructInsert(variables.driveCore, arrEvents[i], ArrayNew(1));
                    }

                    ArrayAppend(variables.driveCore[arrEvents[i]], objController[arrEvents[i]]);
                }
            }
        </cfscript>
    </cffunction>

    <cffunction name="execute" returntype="void">
        <cfset setEvent(getTriggerEvent()) />
        <cfset runEvent() />
        <cfreturn/>
    </cffunction>

    <cffunction name="runEvent" returntype="void" access="private">
        <cfscript>
            var eCurrent = "";
            var iCount = variables.cap;
            var iEvents = 0;
            var x = 0;
            var fExecutor = "";
```

```

do {
    eCurrent = getEvent();

    if (StructKeyExists(variables.driveCore, eCurrent)) {
        iEvents = ArrayLen(variables.driveCore[eCurrent]);

        for (x = 1; x lte iEvents; x = x + 1) {
            fExecutor = variables.driveCore[eCurrent][x];
            fExecutor();
        }
    }

    iCount = iCount - 1;
} while (NOT isEmpty() AND (iCount gte 0));
</cfscript>
<cfreturn/>
</cffunction>

<cffunction name="getEvent">
    <cfset var event = variables.eventCore.pop() >
    <cfreturn event/>
</cffunction>

<cffunction name="getTriggerEvent" returntype="any">
    <cfreturn evaluate(variables.trigger) />
</cffunction>

<cffunction name="setEvent" returntype="void">
    <cfargument name="value" required="true" />
    <cfset variables.eventCore.push(arguments.value) />
    <cfreturn/>
</cffunction>

<cffunction name="isEmpty" access="private">
    <cfreturn variables.eventCore.empty() />
</cffunction>

<cffunction name="clearEvents" returntype="void">
    <cfset variables.eventCore.empty() />
    <cfreturn/>
</cffunction>

</cfcomponent>

```