

Java and ESL coexistence

[Pictures](#) and [Events](#) can contain also [ESL](#) and [JAVA](#) scripts together.

These scripts run in a parallel manner and independently from each other. The initialization parts (between BEGIN and END in ESL, and *onInit()* in Java) are also executed independently on each other. Therefore, if some shared status (e.g. picture variable) is initialized in the initialization part of ESL, this status may not be visible in the initialization part of Java. But the causality of events is kept, i.e. if a picture local variable is set in ESL and consequently RPC is called to Java, the local variable will be already set in Java when executing RPC. In the case that it is necessary to initiate the status in ELS and Java inconsistently and then perform some tasks in ESL, it can be solved in the following manner:

ESL PART

```
BEGIN
; initiate the status in ESL
; ...
CALLJ ... myInit() SYNC ...
; both Java part onInit() and myInit() have already been done
;
;
END
```

JAVA PART

```
public void myInit() {

    // initialization of status in Java

    // ...

}
```

or

ESL PART

```
RPC PROCEDURE DoSomething
; even BEGIN..END part and onInit() and myInit() have already been done
END DoSomething

BEGIN
; initiate status in ESL
; ...
CALLJ ... myInit() ASYNC ...
END
-----
```

JAVA PART

```
public void myInit() {  
  
    // initialization of status in Java  
  
    // ...  
  
    callESLAsync(null, null, 0, "DoSomething", null);  
  
}
```



Related pages:

[Java as a script language](#)
[Event Script Language \(ESL\)](#)