

# GetRPCCallerObject

## %GetRPCCallerObject function

### Function

The function returns a unique identifier (HOBJ) of object of the [Event](#), [Picture](#) type that called currently executed RPC procedure.

### Declaration

```
INT %GetRPCCallerObject()
```

### Parameters

No parameters.

### Return value

The return value is of the *Int* type.

### Description

This function can be used to identify the calling [Event](#) or [Picture](#) in an application that is built on client-server architecture. It eliminates the explicit transmission of process identifier by means of parameters. This example shows the transmission of identifier of calling process by means of parameter and its utilization for the purpose of responding without use of %GetRPCCallerObject and [%GetRPCCallerProcess](#):

### Calling

### Example

```
PROCEDURE ....  
; Complete identification of calling process by means of parameters  
INT _callerProcess, _callerObject  
_callerObject := %GetSelfHBJ(@FALSE)  
_callerProcess := %GetParentProcessHBJ()  
;calling  
CALL [E.Service] Question(_callerProcess, _callerObject) ON SELF.EVH  
; ....  
END ....
```

Handling of calls on the server side:

```
; Handling  
RPC PROCEDURE Question (IN INT _callerProcess, _callerObject)  
; Response!!! ASYNC required, if the Question procedure is called  
synchronously - otherwise DeadLock occurs!!!  
CALL [(_callerObject)] QuestionReply ASYNC ON (_callerProcess)  
END Question
```

By means of %GetRPCCallerProcess, %GetRPCCallerObject:

### Calling

```
PROCEDURE ....  
;calling  
CALL [E.Service] Question ON SELF.EVH  
; ....  
END ....
```

Handling of calls on the server side:

```
RPC PROCEDURE Question
    ; identification of calling process
    INT _callerProcess, _callerObject
    _callerObject := %GetRPCCallerProcess()
    _callerProcess := %GetRPCCallerObject()
    ; Response!!! ASYNC required, if the Question procedure is called
    synchronously - otherwise DeadLock occurs!!!
    IF %GetRPCCallerIsJava() THEN
        CALLJ [(_callerObject)] QuestionReply ASYNC ON (_callerProcess)
    ELSE
        CALL [(_callerObject)] QuestionReply ASYNC ON (_callerProcess)
    ENDIF
END Question
```



**Related pages:**

[ESL extended functions](#)

[Function arguments - types](#)

[CALL action - remote procedure call](#)