XTRA dmmScript for scripting

List of functions

new	XTRA initialisation.
registration	Registration of dmmScript.x32.
runCPP	The function processes the script in C++.
runJS	The function processes the script in JavaScript.
runPas	The function processes the script in Pascal.
runVB	The function processes the script in Visual Basic.
codeError	This event is called in case that code we would like to process returns an error.

new

It is necessary to initiate the dmmScript XTRA before the first use.

Example - Director

```
global scr
openXlib the pathName&"dmmScript.x32"
scr=new(xtra "dmmScript")

If the library dmmXLS.x32 is located in the XTRAS folder it is enough to insert
global xls
xls=new(xtra "dmmXLS")
```

Example - Authorware

```
scr:=NewObject("dmmScript")
```

void=registration(name: string, code:string)

This function has to be called before the first use of the dmmScript immediately after its initialisation. Unless the right registration name and number are inserted, an announcement "this is a demo version" will appear.

Parameters

Type of name is string, for the demo version name = "dmm".
Type of code is string, for the demo version code= "demo". The chain for commercial version is unique.
To register the user will receive parameters code and name.

Example - Director

```
global scr
scr.registration("dmm", "demo")
```

Example - Authorware

```
CallObject(scr; "registration"; "dmm"; "demo")
```

value=runCPP(code: string, returnVar: string)

The function processes the script in C++. It returns a variable defined in the parameter returnVar. A variable of the same name must, of course, also be defined in the script. The function returns such a variable that is defined in the script. If, for example, the variable in script defined in the parameter returnVar is of integer type, then the function returns integer. If the variable in the script is string, the returned value will be string etc.

Parameters

The function has two parameters: code and returnVar. Code is the script in C++ we want to be processed, returnVar is name of the variable in script whose value we want to find.

Example - Director

```
val=scr.runCPP(programCode, "s")
```

Example - Authorware

```
val:=CallObject(scr; "runCPP"; programCode; "s")
```

value=runJS(code: string, returnVar: string)

The function processes the script in JavaScript. It returns a variable defined in the parameter returnVar. A variable of the same name must, of course, also be defined in the script. The function returns such a variable that is defined in the script. If, for example, the variable in script defined in the parameter returnVar is of integer type, then the function returns integer. If the variable in the script is string, the returned value will be string etc.

Parameters

The function has two parameters: code and returnVar. Code is the script in JavaScript we want to be processed, returnVar is name of the variable in script whose value we want to find.

Example - Director

```
val=scr.runJS(programCode, "s")
```

Example - Authorware
```
val:=CallObject(scr; "runJS"; programCode; "s")
```

value=runPas(code: string, returnVar: string)
The function processes the script in Pascal. It returns a variable defined in the parameter returnVar. A variable of the same name must, of course, also be defined in the script. The function returns such a variable that is defined in the script. If, for example, the variable in script defined in the parameter returnVar is of integer type, then the function returns integer. If the variable in the script is string, the returned value will be string etc.

Parameters
The function has two parameters: code and returnVar. Code is the script in Pascal we want to be processed, returnVar is name of the variable in script whose value we want to find.

Example - Director
```
val=scr.runPas(programCode, "s")
```

Example - Authorware
```
val:=CallObject(scr; "runPas"; programCode; "s")
```

value=runVB(code: string, returnVar: string)
The function processes the script in Visual Basic. It returns a variable defined in the parameter returnVar. A variable of the same name must, of course, also be defined in the script. The function returns such a variable that is defined in the script. If, for example, the variable in script defined in the parameter returnVar is of integer type, then the function returns integer. If the variable in the script is string, the returned value will be string etc.

Parameters
The function has two parameters: code and returnVar. Code is the script in Visual Basic we want to be processed, returnVar is name of the variable in script whose value we want to find.

Example - Director
```
val=scr.runVB(programCode, "s")
```

Example - Authorware
```
val:=CallObject(scr; "runVB"; programCode; "s")
```

codeError row, column, err
This event is called in case that code we would like to process returns an error.

Parameters
In the parameters row and column, position of the error in the script represented by row and column number is returned.

Example - Director
```
on codeError row, column, err
    put row, column, err
end
```