Stored Procedures and Macros

Stored procedures and macros allows to reuse written Odysseu Script.

#PROCEDURE

The #PROCEDURE allows to create stored procedures that are saved into the data dictionary and can be reused by any other scripts, see #EXECUTE how to run them.

Usage

Procedures can be compared to stored procedures in a database system or like methods in a programming language. It allows defining a snippet of Odysseus Script that is reusable. The structure for creating a new procedure is as follows:

```
#PROCEDURE <procedure-name>
<variable 1>
<variable 2>
...
<variable n>

BEGIN

    <arbitrary odysseus script commands>

END
```

First, the name, which must be after the #PROCEDURE in the same line, is defined. After that, there might be an ordered list of variables. The order is important! These variables will be used as the parameters. Finally, there can be any Odysseus-Script code including queries etc. between the BEGIN and END. The only exception are "global" variables. Between BEGIN and END, only variables are allowed that are defined before (after the procedure name). You can execute the procedure by using the #EXECUTE command.

It is not possible to install more than one procedure with the same name, so you may delete a procedure by calling #DROPPROCEDURE before.

Example

The example creates a procedure with name "setSomething" and has two parameters (varX and attribute). Remeber, order is important! The procedure calls a #RUNQUERY-Command including both parameters. See #EXECUTE command how this example is used.

```
#PROCEDURE setSomething
varX
attribute
BEGIN
    #RUNQUERY
    SELECT 1 AS a, 2 AS ${attribute}, * FROM bid WHERE b>${varX}
END
```

#EXECUTE

The #EXECUTE command can be used for running installed procedures, which were created by using the #PROCEDURE command.

Usage

Procedures can be compared to stored procedures in a database system or like methods in a programming language. It allows defining a snippet of Odysseus Script that is reusable. Installed procedures can be executed at follows:

```
#EXECUTE <procedure-name>(<variable 1>, <variable 2>, ...,  <variable n>)
```

The usage is similar to function calls in programming languages like Java. After #EXECUTE the name of procedure with a comma-separated list of its parameters. The parameters must be according to the definition of the procedure. If there were two parameters defined by #PROCEDURE, here are also two parameters needed. Notice, the order is important!

You can create the procedure by using the #PROCEDURE command or delete a procedure by calling #DROPPROCEDURE

Example

This example concludes the one from #PROCEDURE. Therefore we have two parameters (varX and attribute). The following executes the "setSomething" procedure and sets varX=1 and attribute=b.

```
#EXECUTE setSomething(1, b)
```

The variables are replaced and the according Odysseus Script of the procedure is executed at this point. Therefore, the following is inserted insted of the #EXECUTE command:

```
    #RUNQUERY
    SELECT 1 AS a, 2 AS b, * FROM bid WHERE b>1
```

Since this snippet is simply inserted, all things of the surrounding Odysseus Script is taken. In our example, this is for example, the choosen #PARSER and #TRANSCFG that are necessary for #RUNQUERY here. Therefore, it is not guaranteed that a procedure is executable of its own.

#DROPPROCEDURE

The command can be used to remove stored procedures, which were created by using the #PROCEDURE (see for more information about procedures) command.

Usage

An installed procedure can be removed by the command followed by the procedure name:

#DROPPROCEDURE<procedure-name>

You can create the procedure by using the #PROCEDURE command or execute a procedure by calling #EXECUTE

Example

This example concludes the one from #PROCEDURE. Therefore, we want to remove the "setSomething" procedure:

```
#DROPPROCEDURE setSomething
```