DB_SET_PROCESS_PARAMS

DB_SET_PROCESS_PARAMS action

Function	This action is used for setting the values of named parameters or deleting the parameters of "context".

Declaration

```
DB_SET_PROCESS_PARAMS paramsRec
```

Parameters

paramsRec	in	Identifier of entire structure. The structured value consists of two columns: • paramName : String(64) • paramValue : String(64)

paramName defines the name of parameter. **paramValue** defines its value. This table shows the permitted combination of values and their interpretation.

Impact on the result	paramName	paramValue
Inserts or updates the parameter value	defined	defined
Deletes the parameter	defined	*(invalid value)*
Ignores (it is not a part of requirements for change)	*(invalid value)*	defined
Ignores (it is not a part of requirements for change)	*(invalid value)*	*(invalid value)*

Warning: If there is the situation that the parameter names are duplicated in one calling of action, the undefined behavior occurs!

Description

The "context" is assigned uniquely to the **group of processes**. The group of processes consists of:

- always the individual process D2000 EventHandler (without using the parameter --batch_mode) or
- in case of the active pictures - a triplet of processes D2000 HI (HIP), **HIS** and **D2000 Server Event Executor** (SEE).

If process **D2000 EventHandler** is started with the parameter *--batch_mode*, "context" is assigned uniquely to a group of events, which are opened with the same instance number (OPENEVENT), not to whole process. The parameter enables the run of several events on the one EventHandler, and each of them can use own "context". In the case of identical instance numbers, the "context" is mutual shared (is common) and the conditions for implementation of database operation are therefore the same.

The context represents the group of uniquely named parameters where every parameter has a value. It is created by first usage of **DB_SET_PROCESS_PARAMS** action. It is handled by D2000 Server. The context is terminated after finishing the last process from the proper group. The contents of context is replicated automatically to all the running D2000 DBManager processes.

Before executing any database command, which was initiated within context, **D2000 DbManager**, inserts its contents into the table *D2000_PROCESS_PARAMS*. It inserts the parameter name into the column *PARAM_NAME* and its value into the column *PARAM_VALUE*. **DbManager** creates the table *D2000_PROCESS_PARAMS* when it is used for the first time.

Note 1

This action can be used to create the parameterized views where the parameters come from the table *D2000_PROCESS_PARAMS*. The table is set automatically by DbManager according to the context in which the database example is used.

Note 2

Database table *D2000_PROCESS_PARAMS* is Global Temporary Table. The sessions see only rows that they wrote there. After deleting a session, the rows, bound to the given session, will be deleted automatically from the table. For more information, as well as DLL, see the documentation for Global Temporary Table.

Example

```
  ; this example validates the result of DB_SET_PROCESS_PARAMS by the
reading of table D2000_PROCESS_PARAMS
 ; insert/update of parameter that is bound to the process (it updates the
table D2000_PROCESS_PARAMS)

 RECORD NOALIAS (SD.Params) _params

 TEXT _cmd ; SQL select
 INT _retCode ; the return value


 _params[1]^PARAM_NAME := "NAME"
 _params[1]^PARAM_VALUE := "SELF.DBM"

 DB_SET_PROCESS_PARAMS _params
 REDIM _params[0]


 _cmd := "SELECT PARAM_NAME, PARAM_VALUE FROM D2000_PROCESS_PARAMS WHERE
PARAM_NAME = 'NAME'"

 ; it reads the table
 SQL_SELECT DBC_TEST_DSN, _retCode, 1000, _cmd BIND _params
 IF _retCode = _ERR_NO_ERROR THEN
  ; the structure _params is filled, it contains one line
 ELSE
  ; an error
 ENDIF
```

Related topics	[All database actions](#)

ⓘ **Related pages:**

[Script actions](#)