

Table of Contents

- EXEC_SP** 2
- Syntax** 2
- Pagination in Queries implemented as stored procedures*** 2
- Examples** 3

[Index](#), [Action](#), [Action List](#), [Process](#), [Query](#), [SQL](#), [v4.8](#)

EXEC_SP

This action allows execution of a database stored procedure.

Syntax

```
EXEC_SP SPName [ OF DatabaseEnvironment ] [ WITH ParamName = ArithmeticExpression () [ INOUT ] | OUT ] ( , ParamName = ArithmeticExpression () [ INOUT ] | OUT ) * ] [ RETURN ObjectName ]
```

where:

- `SPName` is the name of the stored procedure to execute (must be enclosed in apostrophe)
- `DatabaseEnvironment` if present indicates the name of the database environment where the stored procedure is defined (see objects persisted in [external databases](#)). If not present the native AwareIM database is used.
- `ParamName` is the name of the parameter of the stored procedure (if stored procedure requires a parameter. A parameter can be IN, OUT or INOUT. If parameter type is omitted IN is assumed)
- `ArithmeticExpression` an AwareIM expression to initialize input parameters with. If the parameter is of OUT or INOUT type this expression must be the name of an attribute of a business object
- `ObjectName` if the stored procedure returns a result set (the result of the SQL SELECT statement) AwareIM can convert each record to the instance of the specified object. An object that is persisted in the database table of the SELECT statement must be defined. The resulting object instances are either placed in the Context if the EXEC_SP action is used in rules or returned by a query if the action is used in a rule form of the query

Pagination in Queries implemented as stored procedures

In order to support pagination in queries implemented as stored procedures you need to define two parameters for your stored procedure with special names - `LIMIT_START_PARAM` and `LIMIT_SPAN_PARAM`. The former defines the start of the range to be shown by the query and the latter - how many records to return. A stored procedure can then use the values of these parameters to limit returned records. For example. in MySQL you can use the LIMIT keyword. Note that AwareIM will ignore any values for these parameters that you specify when defining a query (but you still need to provide some values to conform with the syntax of the query). For example:

```
EXEC_SP 'Some name' WITH @LIMIT_SPAN_PARAM=0, @LIMIT_START_PARAM=0 RETURN SomeObject
```

At runtime values of zeroes for these parameters will be replaced with proper values that pagination requires.

Examples

```
EXEC_SP 'procAlertGetAll' RETURN Alert
```

This stored procedure does not require any parameters. It selects all alert records in the native database, that AwareIM will automatically convert to instances of the Alert object and put in the context or make available for a query

```
EXEC_SP 'procAlertGet' WITH '@alerID'=1 RETURN Alert
```

This stored procedure returns alert record with id=1

```
EXEC_SP 'procAlertGet' WITH '@alerID'=SPParam.AlertId RETURN Alert
```

This stored procedure returns alert record with id taken from the AlertId attribute of the SPParam object

```
EXEC_SP 'procAlertGetOut' WITH  
'@alerID'=SPParam.AlertId, '@alertName'=SPParam.AlertName OUT
```

This stored procedure returns alert record with id taken from the AlertId attribute of the SPParam object. The name of the alert is then written into the AlertName attribute.

```
EXEC_SP 'proc1' OF SQLServer WITH '@param1'=1, '@param2'=2 RETURN SomeObject
```

This stored procedure returns the specified records from an [external database](#) identified by the name SQLServer.

From:
<http://www.awareim.com/dokuwiki/> - **Documentation**

Permanent link:
http://www.awareim.com/dokuwiki/a_f/a/exec_sp?rev=1761982611

Last update: **2025/11/01 07:36**

