

**NAME**

`SPI_execute_extended` - execute a command with out-of-line parameters

**SYNOPSIS**

```
int SPI_execute_extended(const char *command,
                        const SPIExecuteOptions * options)
```

**DESCRIPTION**

`SPI_execute_extended` executes a command that might include references to externally supplied parameters. The command text refers to a parameter as  $\$n$ , and the `options->params` object (if supplied) provides values and type information for each such symbol. Various execution options can be specified in the `options` struct, too.

The `options->params` object should normally mark each parameter with the `PARAM_FLAG_CONST` flag, since a one-shot plan is always used for the query.

If `options->dest` is not NULL, then result tuples are passed to that object as they are generated by the executor, instead of being accumulated in `SPI_tuptable`. Using a caller-supplied `DestReceiver` object is particularly helpful for queries that might generate many tuples, since the data can be processed on-the-fly instead of being accumulated in memory.

**ARGUMENTS**

```
const char * command
    command string
```

```
const SPIExecuteOptions * options
    struct containing optional arguments
```

Callers should always zero out the entire `options` struct, then fill whichever fields they want to set. This ensures forward compatibility of code, since any fields that are added to the struct in future will be defined to behave backwards-compatibly if they are zero. The currently available `options` fields are:

```
ParamListInfo params
    data structure containing query parameter types and values; NULL if none
```

```
bool read_only
    true for read-only execution
```

```
bool allow_nonatomic
    true allows non-atomic execution of CALL and DO statements (but this field is ignored unless the
```

SPI\_OPT\_NONATOMIC flag was passed to **SPI\_connect\_ext**)

bool *must\_return\_tuples*

if true, raise error if the query is not of a kind that returns tuples (this does not forbid the case where it happens to return zero tuples)

uint64 *tcount*

maximum number of rows to return, or 0 for no limit

DestReceiver \* *dest*

DestReceiver object that will receive any tuples emitted by the query; if NULL, result tuples are accumulated into a *SPI\_tuptable* structure, as in **SPI\_execute**

ResourceOwner *owner*

This field is present for consistency with **SPI\_execute\_plan\_extended**, but it is ignored, since the plan used by **SPI\_execute\_extended** is never saved.

## RETURN VALUE

The return value is the same as for **SPI\_execute**.

When *options->dest* is NULL, *SPI\_processed* and *SPI\_tuptable* are set as in **SPI\_execute**. When *options->dest* is not NULL, *SPI\_processed* is set to zero and *SPI\_tuptable* is set to NULL. If a tuple count is required, the caller's DestReceiver object must calculate it.