NAME

SPI_execute_plan_extended - execute a statement prepared by **SPI_prepare**

SYNOPSIS

int SPI_execute_plan_extended(SPIPlanPtr *plan*,
const SPIExecuteOptions * *options*)

DESCRIPTION

SPI_execute_plan_extended executes a statement prepared by **SPI_prepare** or one of its siblings. This function is equivalent to **SPI_execute_plan**, except that information about the parameter values to be passed to the query is presented differently, and additional execution-controlling options can be passed.

Query parameter values are represented by a ParamListInfo struct, which is convenient for passing down values that are already available in that format. Dynamic parameter sets can also be used, via hook functions specified in ParamListInfo.

Also, instead of always accumulating the result tuples into a *SPI_tuptable* structure, tuples can be passed to a caller-supplied DestReceiver object as they are generated by the executor. This 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

SPIPlanPtr *plan*
prepared statement (returned by **SPI_prepare**)

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_plan**

ResourceOwner *owner*
: The resource owner that will hold a reference count on the plan while it is executed. If NULL, CurrentResourceOwner is used. Ignored for non-saved plans, as SPI does not acquire reference counts on those.

RETURN VALUE

The return value is the same as for **SPI_execute_plan**.

When *options->dest* is NULL, *SPI_processed* and *SPI_tuptable* are set as in **SPI_execute_plan**. 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.