

**NAME**

`SPI_saveplan` - save a prepared statement

**SYNOPSIS**

`SPIPlanPtr SPI_saveplan(SPIPlanPtr plan)`

**DESCRIPTION**

**SPI\_saveplan** copies a passed statement (prepared by **SPI\_prepare**) into memory that will not be freed by **SPI\_finish** nor by the transaction manager, and returns a pointer to the copied statement. This gives you the ability to reuse prepared statements in the subsequent invocations of your C function in the current session.

**ARGUMENTS**

`SPIPlanPtr plan`  
the prepared statement to be saved

**RETURN VALUE**

Pointer to the copied statement; or NULL if unsuccessful. On error, *SPI\_result* is set thus:

**SPI\_ERROR\_ARGUMENT**  
if *plan* is NULL or invalid

**SPI\_ERROR\_UNCONNECTED**  
if called from an unconnected C function

**NOTES**

The originally passed-in statement is not freed, so you might wish to do **SPI\_freeplan** on it to avoid leaking memory until **SPI\_finish**.

In most cases, **SPI\_keepplan** is preferred to this function, since it accomplishes largely the same result without needing to physically copy the prepared statement's data structures.