



# IBEXPERT WHITE PAPER



## Firebird and IBExpert White Paper

# Advanced debugging event block

Fikret Hasovic, February 2022

The *Event Blocks* feature allows you to process certain events within IBExpert.

The latest IBExpert version includes a new feature for advanced debugging, but the blocks have to be added from external templates. These are PSQL Debugger events:

## Before debugging a PSQL object and On Get modified PSQL source

The screenshot shows the 'Edit event blocks' dialog box. At the top, it says 'Double-click or press ENTER to create a new event block'. Below is a table with two columns: 'Event' and 'Assigned block'. The 'Event' column lists various event categories, and the 'Assigned block' column shows which template is assigned to each. The 'PSQL Debugger events' section is highlighted with a blue border, and the 'Debugger :: Before debug a PSQL object' and 'Debugger :: On get modified PSQL source' templates are listed under it. At the bottom of the dialog are 'Create block', 'Cancel', and 'Help' buttons.

Event	Assigned block
IBExpert start/stop	
IBExpert :: After IBExpert starts	
IBExpert :: Before IBExpert stops	
Database connect/disconnect	
Before connect to database	
After database connected	
Before disconnect from database	
After database disconnected	
Metadata Events	
Metadata :: Before metadata change	
Metadata :: After metadata change	
SQL Editor events	
SQL Editor :: On get transaction parameters	
Code Editor events	
Code Editor :: After load from file	
Code Editor :: Before save to file	
Code Editor :: Process the Code Completion list	
Template events	
Templates :: Process general template	
DB object editor events	
Table\view editor :: On compose SELECT statement	
Data grid events	
Data grid :: Before edit a string value	
Data grid :: After executing a query	
Dependencies tree events	
Dependencies :: Before deactivate database objects	
PSQL Debugger events	
Debugger :: Before debug a PSQL object	
Debugger :: On get modified PSQL source	

So, the first block (*Before debugging a PSQL object*) template now looks as follows:

```
execute ibeblock (
    DatabaseID variant comment      'Internal ID (within IBExpert) of
    the database associated to the PSQL debugger',
    ObjectName variant comment      'PSQL object name',
```

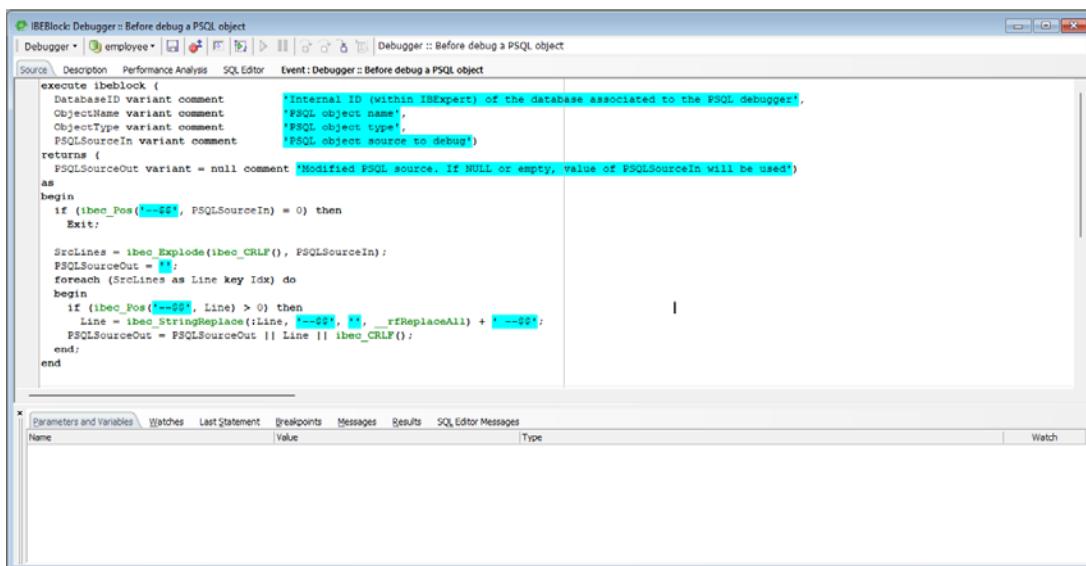


# IBEXPERT WHITE PAPER



```
ObjectType variant comment           'PSQL object type',
PSQLSourceIn variant comment        'PSQL object source to debug'
returns (
    PSQLSourceOut variant = null comment 'Modified PSQL source. If NULL or
empty, value of PSQLSourceIn will be used')
as
begin
    if (ibec_Pos('--$$', PSQLSourceIn) = 0) then
        Exit;

    SrcLines = ibec_Explode(ibec_CRLF(), PSQLSourceIn);
    PSQLSourceOut = '';
    foreach (SrcLines as Line key IIdx) do
    begin
        if (ibec_Pos('--$$', Line) > 0) then
            Line = ibec_StringReplace(:Line, '--$$', '', __rfReplaceAll) + ' --';
    end;
    PSQLSourceOut = PSQLSourceOut || Line || ibec_CRLF();
end;
```



And the second one (*On Get modified PSQL source*) template:

```
execute ibeblock (
    DatabaseID variant comment           'Internal ID (within IBExpert) of
the database associated to the PSQL debugger',
    ObjectName variant comment           'PSQL object name',
    ObjectType variant comment           'PSQL object type',
```

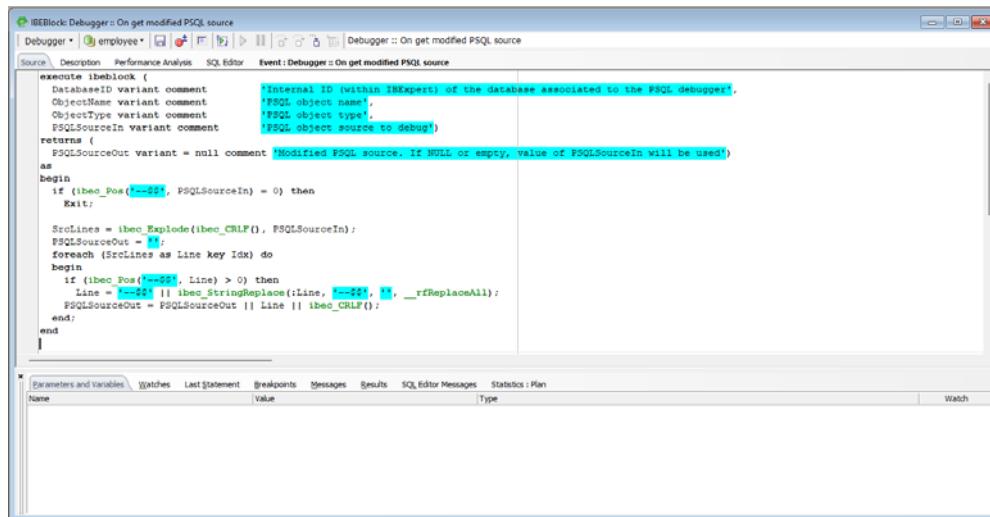


# IBEXPERT WHITE PAPER



```
PSQLSourceIn variant comment           'PSQL object source to debug')
returns (
    PSQLSourceOut variant = null comment 'Modified PSQL source. If NULL or
empty, value of PSQLSourceIn will be used')
as
begin
    if (ibec_Pos('--$$', PSQLSourceIn) = 0) then
        Exit;

    SrcLines = ibec_Explode(ibec_CRLF(), PSQLSourceIn);
    PSQLSourceOut = '';
    foreach (SrcLines as Line key Idx) do
    begin
        if (ibec_Pos('--$$', Line) > 0) then
            Line = '--$$' || ibec_StringReplace(:Line, '--$$', '',
__rfReplaceAll);
        PSQLSourceOut = PSQLSourceOut || Line || ibec_CRLF();
    end;
end
```



In practice, this means that you can use the new method of commenting while debugging. To comment you should use following string: `--$$.` The comment will be removed while in the debugger. For example:

The original source would be something like:

```
create procedure
returns (id bigint)
as
begin
    for
        select id
```



# IBEXPERT WHITE PAPER



```
    from customer
--$$ where id=123
    into id
do
suspend;
end
```

The *before event* block will change the code to:

```
create procedure
returns (id bigint)
as
begin
for
    select id
    from customer
    where id=123  --$$
    into id
do
suspend;
end
```

and the *after event* block changes the code back before compilation:

```
create procedure
returns (id bigint)
as
begin
for
    select id
    from customer
--$$ where id=123
    into id
do
suspend;
end
```

After that, only when the debugger code was changed and the user confirms the changes to be used for the stored procedure.