ibec_PrepareStatement

Prepares a DML statement and returns preparation report or raises an error.

Syntax

```
ibec_PrepareStatement(Connection : variant; Statement : string; Options :
string; RaiseError : boolean) : string;
```

Connection	Active connection object.
Statement	Source of DML statement.
Options	Options string. Possible options are:
Plan	Include statement plan, if available, into the result report
ExplainPlan	Include statement explain plan, if available, into the result report. For Firebird 3 and above.
Columns	Include information about statement colums: datatype, size, precision etc.
RaiseError	If RaiseError=TRUE and preparation causes an error, an exception will be generated.
If RaiseError=FALSE and preparation causes an error its message, SQL code and IB error code will be included into the result report.	If RaiseError=FALSE and preparation causes an error its message, SQL code and IB error code will be included into the result report.

Example 1

Preparation of invalid statement without raising of error:

```
execute ibeblock
as
begin
  db = ibec_GetDefaultConnection();

  var_sql = 'selects * from rdb$relations';
  var_res = ibec_PrepareStatement(db, var_sql, '', FALSE);
end


Output (JSON):

{
  "Statement": "selects * from rdb$relations\r\n",
  "Error": {
    "SQLCode": -104,
    "IBErrorCode": 335544569,
    "Message": "can't format message 13:896 -- message system code
-4.\r\nDynamic
```

```
                    SQL Error.\r\nSQL error code = -104.\r\nToken unknown -
line 1,
                    column 1.\r\nselects.\r\n"
    }
 }
```

Example 2

Preparation of invalid statement with raising of error:

```
execute ibeblock
as
begin
  db =  ibec_GetDefaultConnection();

  var_sql = 'selects * from rdb$relations';
  try
    var_res = ibec_PrepareStatement(db, var_sql, '', TRUE);
  except
    sMessage = ibec_err_Message();
    ibec_ShowMessage(sMessage);
  end
end
```

Example 3

Preparation of valid statement:

```
execute ibeblock
as
begin
  db =  ibec_GetDefaultConnection();

  var_sql = 'select rdb$relation_name, rdb$relation_id, rdb$description
from rdb$relations';
  var_res = ibec_PrepareStatement(db, var_sql, 'Plan; ExplainPlan;
Columns', FALSE);
  ibec_SaveToFile('D:\temp\prepare.json', var_res, __stfOverwrite);
end
```

```
Output (JSON):

 {
   "Statement": "select rdb$relation_name, rdb$relation_id, rdb$description
from rdb$relations\r\n",
   "SQLType": "1 (Select)",
   "Plan": "PLAN (RDB$RELATIONS NATURAL)",
```

```
    "ExplainPlan": "Select Expression\n     -> Table \"RDB$RELATIONS\" Full
Scan",
    "Columns": [
      {
        "XSQLVAR.Name": "RDB$RELATION_NAME",
        "XSQLVAR.AliasName": "RDB$RELATION_NAME",
        "XSQLVAR.RelationName": "RDB$RELATIONS",
        "XSQLVAR.Type": 453,
        "XSQLVAR.SubType": 4,
        "XSQLVAR.Scale": 0,
        "XSQLVAR.Length": 124,
        "Type": "CHAR(31)"
      },
      {
        "XSQLVAR.Name": "RDB$RELATION_ID",
        "XSQLVAR.AliasName": "RDB$RELATION_ID",
        "XSQLVAR.RelationName": "RDB$RELATIONS",
        "XSQLVAR.Type": 501,
        "XSQLVAR.SubType": 0,
        "XSQLVAR.Scale": 0,
        "XSQLVAR.Length": 2,
        "Type": "SMALLINT"
      },
      {
        "XSQLVAR.Name": "RDB$DESCRIPTION",
        "XSQLVAR.AliasName": "RDB$DESCRIPTION",
        "XSQLVAR.RelationName": "RDB$RELATIONS",
        "XSQLVAR.Type": 521,
        "XSQLVAR.SubType": 1,
        "XSQLVAR.Scale": 4,
        "XSQLVAR.Length": 8,
        "Type": "BLOB SUB_TYPE 1"
      }
    ]
  }
```